Steven A. Frank

Control Theory Tutorial

Basic Concepts Illustrated by Software Examples

EXTRAS ONLINE

Springer Open

SpringerBriefs in Applied Sciences and Technology

SpringerBriefs present concise summaries of cutting-edge research and practical applications across a wide spectrum of fields. Featuring compact volumes of 50–125 pages, the series covers a range of content from professional to academic.

Typical publications can be:

- A timely report of state-of-the art methods
- An introduction to or a manual for the application of mathematical or computer techniques
- A bridge between new research results, as published in journal articles
- A snapshot of a hot or emerging topic
- An in-depth case study
- A presentation of core concepts that students must understand in order to make independent contributions

SpringerBriefs are characterized by fast, global electronic dissemination, standard publishing contracts, standardized manuscript preparation and formatting guidelines, and expedited production schedules.

On the one hand, **SpringerBriefs in Applied Sciences and Technology** are devoted to the publication of fundamentals and applications within the different classical engineering disciplines as well as in interdisciplinary fields that recently emerged between these areas. On the other hand, as the boundary separating fundamental research and applied technology is more and more dissolving, this series is particularly open to trans-disciplinary topics between fundamental science and engineering.

Indexed by EI-Compendex, SCOPUS and Springerlink.

More information about this series at http://www.springer.com/series/8884

Steven A. Frank

Control Theory Tutorial

Basic Concepts Illustrated by Software
Examples

Springer Open

Steven A. Frank
Department of Ecology and Evolutionary
  Biology
University of California, Irvine
Irvine, CA
USA

Précis

This book introduces the basic principles of control theory in a concise self-study tutorial. The chapters build the foundation of control systems design based on feedback, robustness, tradeoffs, and optimization. The approach focuses on how to think clearly about control and why the key principles are important. Each principle is illustrated with examples and graphics developed by software coded in Wolfram Mathematica. All of the software is freely available for download. The software provides the starting point for further exploration of the concepts and for development of new theoretical studies and applications.

Preface

I study how natural biological processes shape the design of organisms. Like many biologists, I have often turned to the rich theory of engineering feedback control to gain insight into biology.

The task of learning control theory for a biologist or for an outsider from another scientific field is not easy. I read and reread the classic introductory texts of control theory. I learned the basic principles and gained the ability to analyze simple models of control. The core of the engineering theory shares many features with my own closest interests in design tradeoffs in biology. How much cost is it worth paying to enhance performance? What is the most efficient investment in improved design given the inherent limitation on time, energy, and other resources?

Yet, for all of the conceptual similarities to my own research and for all of my hours of study with the classic introductory texts, I knew that I had not mastered the broad principles of engineering control theory design. How should I think simply and clearly about a basic control theory principle such as integral control in terms of how a biological system actually builds an error-correcting feedback loop? What is the relation between various adaptive engineering control systems and the ways in which organisms build hard-wired versus flexible control responses? How do the classic cost-benefit analyses of engineering quadratic control models relate to the commonly used notions of costs and benefits in models of organismal design?

After several years of minor raiding around the periphery of engineering control theory, I decided it was time to settle down and make a carefully planned attack. I lined up the classic texts, from the basic introductions to the more advanced treatises on nonlinear control, adaptive control, model predictive control, modern robust analysis, and the various metrics used to analyze uncertainty. I could already solve a wide range of problems, but I had never fully internalized the basic principles that unified the subject in a simple and natural way.

This book is the tutorial that I developed for myself. This tutorial can guide you toward broad understanding of the principles of control in a way that cannot be obtained from the standard introductory books. Those classic texts are brilliant compilations of knowledge with excellent drills to improve technical skill. But those texts cannot teach you to understand the principles of control, how to

internalize the concepts and make them your own. You must ultimately learn to think simply and clearly about problems of control and how such problems relate to the broad corpus of existing knowledge.

At every stage of learning, this tutorial provides the next natural step to move ahead. I present each step in the quickest and most illustrative manner. If that quick step works for you, then you can move along. If not, then you should turn to the broad resources provided by the classic texts. In this way, you can build your understanding rapidly, with emphasis on how the pieces fit together to make a rich and beautiful conceptual whole. Throughout your study, you can take advantage of other sources to fill in technical gaps, practical exercises, and basic principles of applied mathematics.

You will have to build your own course of study, which can be challenging. But with this tutorial guide, you can do it with the confidence that you are working toward the broad conceptual understanding that can be applied to a wide range of real-world problems. Although the size of this tutorial guide is small, it will lead you toward the key concepts in standard first courses plus many of the principles in the next tier of advanced topics. For scientists outside of engineering, I cannot think of another source that can guide your study in such a simple and direct way. For engineering students, this tutorial supplements the usual courses and books to unify the conceptual understanding of the individual tools and skills that you learn in your routine studies.

This tutorial is built around an extensive core of software tools and examples. I designed that software to illustrate fundamental concepts, to teach you how to do analyses of your own problems, and to provide tools that can be used to develop your own research projects. I provide all of the software code used to analyze the examples in the text and to generate the figures that illustrate the concepts.

The software is written in Wolfram Mathematica. I used Mathematica rather than the standard MATLAB tools commonly used in engineering courses. Those two systems are similar for analyzing numerical problems. However, Mathematica provides much richer tools for symbolic analysis and for graphic presentation of complex results from numerical analysis. The symbolic tools are particularly valuable, because the Mathematica code provides clear documentation of assumptions and mathematical analysis along with the series of steps used in derivations. The symbolic analysis also allows easy coupling of mathematical derivations to numerical examples and graphical illustrations. All of the software code used in this tutorial is freely available at http://extras.springer.com/2018/978-3-319-91707-8.

Irvine, USA   Steven A. Frank
March 2018

Contents

Chapter 1
Introduction

I introduce the basic principles of control theory in a concise self-study guide. I wrote this guide because I could not find a simple, brief introduction to the foundational concepts. I needed to understand those key concepts before I could read the standard introductory texts on control or read the more advanced literature. Ultimately, I wanted to achieve sufficient understanding so that I could develop my own line of research on control in biological systems.

This tutorial does not replicate the many excellent introductory texts on control theory. Instead, I present each key principle in a simple and natural progression through the subject.

The principles build on each other to fill out the basic foundation. I leave all the detail to those excellent texts and instead focus on how to think clearly about control. I emphasize why the key principles are important, and how to make them your own to provide a basis on which to develop your own understanding.

I illustrate each principle with examples and graphics that highlight key aspects. I include, in a freely available file, all of the Wolfram Mathematica software code that I used to develop the examples and graphics (see Preface). The code provides the starting point for your own exploration of the concepts and the subsequent development of your own theoretical studies and applications.

1.1 Control Systems and Design

An incoming gust of wind tips a plane. The plane's sensors measure orientation. The measured orientation feeds into the plane's control systems, which send signals to the plane's mechanical components. The mechanics reorient the plane.

An organism's sensors transform light and temperature into chemical signals. Those chemical signals become inputs for further chemical reactions. The chain of chemical reactions feeds into physical systems that regulate motion.

How should components be designed to modulate system response? Different goals lead to design tradeoffs. For example, a system that responds rapidly to changing input signals may be prone to overshooting design targets. The tradeoff between performance and stability forms one key dimension of design.

Control theory provides rich insights into the inevitable tradeoffs in design. Biologists have long recognized the analogies between engineering design and the analysis of biological systems. Biology is, in essence, the science of reverse engineering the design of organisms.

1.2   Overview

I emphasize the broad themes of feedback, robustness, design tradeoffs, and optimization. I weave those themes through the three parts of the presentation.

1.2.1   Part I: Basic Principles

The first part develops the basic principles of dynamics and control. This part begins with alternative ways in which to study dynamics. A system changes over time, the standard description of dynamics. One can often describe changes over time as a combination of the different frequencies at which those changes occur. The duality between temporal and frequency perspectives sets the classical perspective in the study of control.

The first part continues by applying the tools of temporal and frequency analysis to basic control structures. Open-loop control directly alters how a system transforms inputs into outputs. Prior knowledge of the system's intrinsic dynamics allows one to design a control process that modulates the input–output relation to meet one's goals.

By contrast, closed-loop feedback control allows a system to correct for lack of complete knowledge about intrinsic system dynamics and for unpredictable perturbations to the system. Feedback alters the input to be the error difference between the system's output and the system's desired target output.

By feeding back the error into the system, one can modulate the process to move in the direction that reduces error. Such self-correction by feedback is the single greatest principle of design in both human-engineered systems and naturally evolved biological systems.

I present a full example of feedback control. I emphasize the classic proportional, integral, derivative (PID) controller. A controller is a designed component of the system that modulates the system's intrinsic input–output response dynamics.

In a PID controller, the proportional component reduces or amplifies an input signal to improve the way in which feedback drives a system toward its target. The integral component strengthens error correction when moving toward a fixed target value. The derivative component anticipates how the target moves, providing a more rapid system response to changing conditions.

The PID example illustrates how to use the basic tools of control analysis and design, including the frequency interpretation of dynamics. PID control also introduces key tradeoffs in design. For example, a more rapid response toward the target setpoint often makes a system more susceptible to perturbations and more likely to become unstable.

This first part concludes by introducing essential measures of performance and robustness. Performance can be measured by how quickly a system moves toward its target or, over time, how far the system tends to be from its target. The cost of driving a system toward its target is also a measurable aspect of performance. Robustness can be measured by how likely it is that a system becomes unstable or how sensitive a system is to perturbations. With explicit measures of performance and robustness, one can choose designs that optimally balance tradeoffs.

1.2.2 Part II: Design Tradeoffs

The second part applies measures of performance and robustness to analyze tradeoffs in various design scenarios.

Regulation concerns how quickly a system moves toward a fixed setpoint. I present techniques that optimize controllers for regulation. *Optimal* means the best balance between design tradeoffs. One finds an optimum by minimizing a cost function that combines the various quantitative measures of performance and robustness.

Stabilization considers controller design for robust stability. A robust system maintains its stability even when the intrinsic system dynamics differ significantly from that assumed during analysis. Equivalently, the system maintains stability if the intrinsic dynamics change or if the system experiences various unpredictable perturbations. Changes in system dynamics or unpredicted perturbations can be thought of as uncertainties in intrinsic dynamics.

The stabilization chapter presents a measure of system stability when a controller modulates intrinsic system dynamics. The stability measure provides insight into the set of uncertainties for which the system will remain stable. The stability analysis is based on a measure of the distance between dynamical systems, a powerful way in which to compare performance and robustness between systems.

Tracking concerns the ability of a system to follow a changing environmental setpoint. For example, a system may benefit by altering its response as the environmental temperature changes. How closely can the system track the optimal response to the changing environmental input? Once again, the analysis of performance and robustness may be developed by considering explicit measures of system characteristics. With explicit measures, one can analyze the tradeoffs between competing goals and how alternative assumptions lead to alternative optimal designs.

All of these topics build on the essential benefits of feedback control. The particular information that can be measured and used for feedback plays a key role in control design.

1.2.3 Part III: Common Challenges

The third part presents challenges in control design. Challenges include nonlinearity and uncertainty of system dynamics.

Classical control theory assumes linear dynamics, whereas essentially all processes are nonlinear. One defense of linear theory is that it often works for real problems. Feedback provides powerful error correction, often compensating for unknown nonlinearities. Robust linear design methods gracefully handle uncertainties in system dynamics, including nonlinearities.

One can also consider the nonlinearity explicitly. With assumptions about the form of nonlinearity, one can develop designs for nonlinear control.

Other general design approaches work well for uncertainties in intrinsic system dynamics, including nonlinearity. Adaptive control adjusts estimates for the unknown parameters of intrinsic system dynamics. Feedback gives a measure of error in the current parameter estimates. That error is used to learn better parameter values. Adaptive control can often be used to adjust a controller with respect to nonlinear intrinsic dynamics.

Model predictive control uses the current system state and extrinsic inputs to calculate an optimal sequence of future control steps. Those future control steps ideally move the system toward the desired trajectory at the lowest possible cost. At each control point in time, the first control step in the ideal sequence is applied. Then, at the next update, the ideal control steps are recalculated, and the first new step is applied.

By using multiple lines of information and recalculating the optimal response, the system corrects for perturbations and for uncertainties in system dynamics. Those uncertainties can include nonlinearities, providing another strong approach for nonlinear control.

Part I
Basic Principles

Chapter 2
Control Theory Dynamics

The mathematics of classical control theory depends on linear ordinary differential equations, which commonly arise in all scientific disciplines. Control theory emphasizes a powerful Laplace transform expression of linear differential equations. The Laplace expression may be less familiar in particular disciplines, such as theoretical biology.

2.1 Transfer Functions and State Space

Here, I show how and why control applications use the Laplace form. I recommend an introductory text on control theory for additional background and many example applications (e.g., Åström and Murray 2008; Ogata 2009; Dorf and Bishop 2016).

Suppose we have a process, P, that transforms a command input, u, into an output, y. Figure 2.1a shows the input–output flow. Typically, we write the process as a differential equation, for example

$$\ddot{x} + a_1\dot{x} + a_2 x = \dot{u} + bu, \tag{2.1}$$

in which $x(t)$ is an internal state variable of the process that depends on time, $u(t)$ is the forcing command input signal, and overdots denote derivatives with respect to time. Here, for simplicity, we let the output be equivalent to the internal state, $y \equiv x$.

The dynamics of the input signal, u, may be described by another differential equation, driven by reference input, r (Fig. 2.1b). Mathematically, there is no problem cascading sequences of differential equations in this manner. However, the rapid growth of various symbols and interactions make such cascades of differential equations difficult to analyze and impossible to understand intuitively.

(a)

$U(s) \xrightarrow{\quad u \quad} \boxed{P(s)} \xrightarrow{\quad y \quad} Y(s)$

(b)

$\xrightarrow{\ r\ } \boxed{C(s)} \xrightarrow{\ u\ } \boxed{P(s)} \xrightarrow{\ y\ }$

(c)

$\xrightarrow{\ r\ } \bigcirc \xrightarrow{\ e\ } \boxed{C(s)} \xrightarrow{\ u\ } \boxed{P(s)} \xrightarrow{\ y\ }$

$\boxed{-1}$
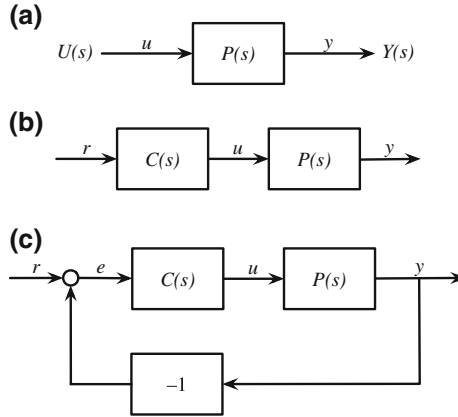
Fig. 2.1 Basic process and control flow. **a** The input–output flow in Eq. 2.2. The input, $U(s)$, is itself a transfer function. However, for convenience in diagramming, lowercase letters are typically used along pathways to denote inputs and outputs. For example, in **a**, u can be used in place of $U(s)$. In **b**, only lowercase letters are used for inputs and outputs. Panel **b** illustrates the input–output flow of Eq. 2.3. These diagrams represent open-loop pathways because no closed-loop feedback pathway sends a downstream output back as an input to an earlier step. **c** A basic closed-loop process and control flow with negative feedback. The circle between r and e denotes addition of the inputs to produce the output. In this figure, $e = r - y$

We can use a much simpler way to trace input–output pathways through a system. If the dynamics of P follow Eq. 2.1, we can transform P from an expression of temporal dynamics in the variable t to an expression in the complex Laplace variable s as

$$P(s) = \frac{Y(s)}{U(s)} = \frac{s+b}{s^2 + a_1 s + a_2}. \tag{2.2}$$

The numerator simply uses the coefficients of the differential equation in u from the right side of Eq. 2.1 to make a polynomial in s. Similarly, the denominator uses the coefficients of the differential equation in x from the left side of Eq. 2.1 to make a polynomial in s. The eigenvalues for the process, P, are the roots of s for the polynomial in the denominator. Control theory refers to the eigenvalues as the *poles* of the system.

From this equation and the matching picture in Fig. 2.1, we may write $Y(s) = U(s)P(s)$. In words, the output signal, $Y(s)$, is the input signal, $U(s)$, multiplied by the transformation of the signal by the process, $P(s)$. Because $P(s)$ multiplies the signal, we may think of $P(s)$ as the signal gain, the ratio of output to input, Y/U. The signal gain is zero at the roots of the numerator's polynomial in s. Control theory refers to those numerator roots as the *zeros* of the system.

The simple multiplication of the signal by a process means that we can easily cascade multiple input–output processes. For example, Fig. 2.1b shows a system with extended input processing. The cascade begins with an initial reference input, r, which is transformed into the command input, u, by a preprocessing controller, C, and then finally into the output, y, by the intrinsic process, P. The input–output calculation for the entire cascade follows easily by noting that $C(s) = U(s)/R(s)$, yielding

$$Y(s) = R(s)C(s)P(s) = R(s)\frac{U(s)}{R(s)}\frac{Y(s)}{U(s)}. \tag{2.3}$$

These functions of s are called *transfer functions*.

Each transfer function in a cascade can express any general system of ordinary linear differential equations for vectors of state variables, x, and inputs, u, with dynamics given by

$$x^{(n)} + a_1 x^{(n-1)} + \cdots + a_{n-1}x^{(1)} + a_n x$$
$$= b_0 u^{(m)} + b_1 u^{(m-1)} + \cdots + b_{m-1}u^{(1)} + b_m u, \tag{2.4}$$

in which parenthetical superscripts denote the order of differentiation. By analogy with Eq. 2.2, the associated general expression for transfer functions is

$$P(s) = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_{m-1}s + b_m}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n}. \tag{2.5}$$

The actual biological or physical process does not have to include higher-order derivatives. Instead, the dynamics of Eq. 2.4 and its associated transfer function can always be expressed by a system of first-order processes of the form

$$\dot{x}_i = \sum_j a_{ij}x_j + \sum_j b_{ij}u_j, \tag{2.6}$$

which allows for multiple inputs, u_j. This system describes the first-order rate of change in the state variables, $\dot{x}_i$, in terms of the current states and inputs. This state-space description for the dynamics is usually written in vector notation as

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$
$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du},$$

which potentially has multiple inputs and outputs, $\mathbf{u}$ and $\mathbf{y}$.

For example, the single input–output dynamics in Eq. 2.1 translate into the state-space model

$$\dot{x}_1 = -a_2 x_2 + bu$$
$$\dot{x}_2 = x_1 - a_1 x_2 + u$$
$$y = x_2,$$

in which the rates of change in the states depend only on the current states and the current input.

2.2   Nonlinearity and Other Problems

Classical control theory focuses on transfer functions. Those functions apply only to linear, time-invariant dynamics. By contrast, state-space models can be extended to any type of nonlinear, time-varying process.

Real systems are typically nonlinear. Nonetheless, four reasons justify the study of linear theory.

First, linear analysis clarifies fundamental principles of dynamics and control. For example, feedback often leads to complex, nonintuitive pathways of causation. Linear analysis has clarified the costs and benefits of feedback in terms of trade-offs between performance, stability, and robustness. Those principles carry over to nonlinear systems, although the quantitative details may differ.

Second, many insights into nonlinear aspects of control come from linear theory (Isidori 1995; Khalil 2002; Astolfi et al. 2008). In addition to feedback, other principles include how to filter out disturbances at particular frequencies, how time delays alter dynamics and the potential for control, how to track external setpoints, and how to evaluate the costs and benefits of adding sensors to monitor state and adjust dynamics.

Third, linear theory includes methods to analyze departures from model assumptions. Those linear methods of robustness often apply to nonlinear departures from assumed linearity. One can often analyze the bounds on a system's performance, stability, and robustness to specific types of nonlinear dynamics.

Fourth, analysis of particular nonlinear systems often comes down to studying an approximately linearized version of the system. If the system state remains near an equilibrium point, then the system will be approximately linear near that point. If the system varies more widely, one can sometimes consider a series of changing linear models that characterize the system in each region. Alternatively, a rescaling of a nonlinear system may transform the dynamics into a nearly linear system.

Given a particular nonlinear system, one can always simulate the dynamics explicitly. The methods one uses to understand and to control a simulated system arise mostly from the core linear theory and from the ways that particular nonlinearities depart from that core theory.

2.3 Exponential Decay and Oscillations

Two simple examples illustrate the match between standard models of dynamics and the transfer function expressions. First, the simplest first-order differential equation in $x(t)$ forced by the input $u(t)$, with initial condition $x(0) = 0$, is given by

$$\dot{x} + ax = u, \tag{2.7}$$

which has the solution

$$x(t) = \int_0^t e^{-a\tau} u(t - \tau) d\tau. \tag{2.8}$$

This process describes how x accumulates over time, as inputs arrive at each time point with intensity u, and x decays at rate a.

If the input into this system is the impulse or Dirac delta function, $u(t)dt = 1$ at $t = 0$ and $u(t) = 0$ for all other times, then

$$x(t) = e^{-at}.$$

If the input is the unit step function, $u(t) = 1$ for $t \geq 0$ and $u(t) = 0$ for $t < 0$, then

$$x(t) = \frac{1}{a} \left(1 - e^{-at}\right).$$

Many processes follow the basic exponential decay in Eq. 2.8. For example, a quantity u of a molecule may arrive in a compartment at each point in time and then decay at rate a within the compartment. At any time, the total amount of the molecule in the compartment is the sum of the amounts that arrived at each time in the past, $u(t - \tau)$, weighted by the fraction that remains after decay, $e^{-a\tau}$.

The process in Eq. 2.7 corresponds exactly to the transfer function

$$P(s) = \frac{1}{s + a}, \tag{2.9}$$

in which the output is equivalent to the internal state, $y \equiv x$.

In the second example, an intrinsic process may oscillate at a particular frequency, ω_0, described by

$$\ddot{x} + \omega_0^2 x = u.$$

This system produces output $x = \sin(\omega_0 t)$ for $u = 0$ and an initial condition along the sine curve. The corresponding transfer function is

$$P(s) = \frac{\omega_0}{s^2 + \omega_0^2}.$$

We can combine processes by simply multiplying the transfer functions. For example, suppose we have an intrinsic exponential decay process, $P(s)$, that is driven by oscillating inputs, $U(s)$. That combination produces an output

$$Y(s) = U(s)P(s) = \frac{\omega_0}{(s + a)(s^2 + \omega_0^2)}, \tag{2.10}$$

which describes a third-order differential equation, because the polynomial of s in the denominator has a highest power of three.

We could have easily obtained that third-order process by combining the two systems of differential equations given above. However, when systems include many processes in cascades, including feedback loops, it becomes difficult to combine the differential equations into very high-order systems. Multiplying the transfer functions through the system cascade remains easy. That advantage was nicely summarized by Bode (1964), one of the founders of classical control theory

> The typical regulator system can frequently be described, in essentials, by differential equations of no more than perhaps the second, third or fourth order. … In contrast, the order of the set of differential equations describing the typical negative feedback amplifier used in telephony is likely to be very much greater. As a matter of idle curiosity, I once counted to find out what the order of the set of equations in an amplifier I had just designed would have been, if I had worked with the differential equations directly. It turned out to be 55.

2.4 Frequency, Gain, and Phase

How do systems perform when parameters vary or when there are external environmental perturbations? We can analyze robustness by using the differential equations to calculate the dynamics for many combinations of parameters and perturbations. However, such calculations are tedious and difficult to evaluate for more than a couple of parameters. Using transfer functions, we can study a wide range of conditions by evaluating a function's output response to various inputs.

This chapter uses the Bode plot method. That method provides an easy and rapid way in which to analyze a system over various inputs. We can apply this method to individual transfer functions or to cascades of transfer functions that comprise entire systems.

This section illustrates the method with an example. The following section describes the general concepts and benefits.

Consider the transfer function

$$G(s) = \frac{a}{s + a}, \tag{2.11}$$

which matches the function for exponential decay in Eq. 2.9. Here, I multiplied the function by a so that the value would be one when $s = 0$.
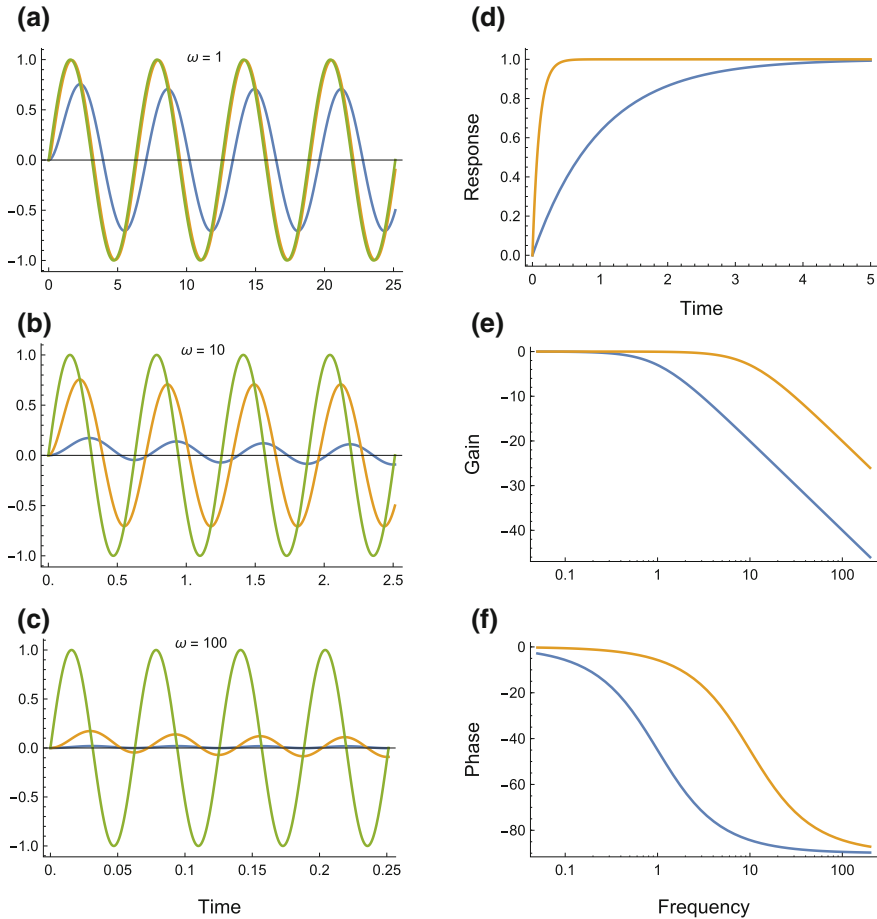
Fig. 2.2 Dynamics, gain, and phase of the low-pass filter in Eq. 2.11 in response to sine wave inputs at varying frequencies, ω. Details provided in the text. **a–c** Dynamics given by a multiplied by the transfer function on the right-hand side of Eq. 2.10. **d** Response of Eq. 2.11 to unit step input. **e** The scaling of the Bode gain plot is $20 \log_{10}(\text{gain})$. That scaling arises from the relation between the magnitude, $M = |G(j\omega)|$, and power, $P = M^2$, of a signal at a particular frequency, ω, or equivalently $M = \sqrt{P}$. If we consider gain as the magnitude of the output signal, then the scale for the gain is given as $20 \log_{10}(\sqrt{P}) = 10 \log_{10}(P)$, the standard decibel scaling for the relative power of a signal. **f** Bode phase plot

We can learn about a system by studying how it responds to different kinds of fluctuating environmental inputs. In particular, how does a system respond to different frequencies of sine wave inputs?

Figure 2.2 shows the response of the transfer function in Eq. 2.11 to sine wave inputs of frequency, ω. The left column of panels illustrates the fluctuating output in response to the green sine wave input. The blue (slow) and gold (fast) responses

correspond to parameter values in Eq. 2.11 of $a = 1$ and $a = 10$. All calculations and plots in this book are available in the accompanying Mathematica code (Wolfram Research 2017) at the site listed in the Preface.

In the top-left panel, at input frequency $\omega = 1$, the fast (gold) response output closely tracks the input. The slow (blue) response reduces the input by $\sqrt{2} \approx 0.7$. This output–input ratio is called the transfer function's *gain*. The slow response output also lags the input by approximately 0.11 of one complete sine wave cycle of $2\pi = 6.28$ radians, thus the shift to the right of $0.11 \times 6.28 \approx 0.7$ radians along the x-axis.

We may also consider the lagging shift in angular units, in which 2π radians is equivalent to $360°$. The lag in angular units is called the *phase*. In this case, the phase is written as $-0.11 \times 360° \approx -40°$, in which the negative sign refers to a lagging response.

A transfer function always transforms a sine wave input into a sine wave output modulated by the gain and phase. Thus, the values of gain and phase completely describe the transfer function response.

Figure 2.2b shows the same process but driven at a higher input frequency of $\omega = 10$. The fast response is equivalent to the slow response of the upper panel. The slow response has been reduced to a gain of approximately 0.1, with a phase of approximately $-80°$. At the higher frequency of $\omega = 100$ in the bottom panel, the fast response again matches the slow response of the panel above, and the slow response's gain is reduced to approximately 0.01.

Both the slow and fast transfer functions pass low-frequency inputs into nearly unchanged outputs. At higher frequencies, they filter the inputs to produce greatly reduced, phase-shifted outputs. The transfer function form of Eq. 2.11 is therefore called a low-pass filter, passing low frequencies and blocking high frequencies. The two filters in this example differ in the frequencies at which they switch from passing low-frequency inputs to blocking high-frequency inputs.

2.5   Bode Plots of Gain and Phase

A Bode plot shows a transfer function's gain and phase at various input frequencies. The Bode gain plot in Fig. 2.2e presents the gain on a log scale, so that a value of zero corresponds to a gain of one, $\log(1) = 0$.

For the system with the slower response, $a = 1$ in blue, the gain is nearly one for frequencies less than a and then drops off quickly for frequencies greater than a. Similarly, the system with faster response, $a = 10$, transitions from a system that passes low frequencies to one that blocks high frequencies at a point near its a value. Figure 2.2f shows the phase changes for these two low-pass filters. The slower blue system begins to lag at lower input frequencies.

Low-pass filters are very important because low-frequency inputs are often external signals that the system benefits by tracking, whereas high-frequency inputs are often noisy disturbances that the system benefits by ignoring.

In engineering, a designer can attach a low-pass filter with a particular transition parameter a to obtain the benefits of filtering an input signal. In biology, natural selection must often favor appending biochemical processes or physical responses that act as low-pass filters. In this example, the low-pass filter is simply a basic exponential decay process.

Figure 2.2d shows a key tradeoff between the fast and slow responses. In that panel, the system input is increased in a step from zero to one at time zero. The fast system responds quickly by increasing its state to a matching value of one, whereas the slow system takes much longer to increase to a matching value. Thus, the fast system may benefit from its quick response to environmental changes, but it may lose by its greater sensitivity to high-frequency noise. That tradeoff between responsiveness and noise rejection forms a common theme in the overall performance of systems.

To make the Bode plot, we must calculate the gain and phase of a transfer function's response to a sinusoidal input of frequency ω. Most control theory textbooks show the details (e.g.,Ogata 2009). Here, I briefly describe the calculations, which will be helpful later.

Transfer functions express linear dynamical systems in terms of the complex Laplace variable $s = \sigma + j\omega$. I use j for the imaginary number to match the control theory literature.

The gain of a transfer function describes how much the function multiplies its input to produce its output. The gain of a transfer function $G(s)$ varies with the input value, s. For complex-valued numbers, we use magnitudes to analyze gain, in which the magnitude of a complex value is $|s| = \sqrt{\sigma^2 + \omega^2}$.

It turns out that the gain of a transfer function in response to a sinusoidal input at frequency ω is simply $|G(j\omega)|$, the magnitude of the transfer function at $s = j\omega$. The phase angle is the arctangent of the ratio of the imaginary to the real parts of $G(j\omega)$.

For the exponential decay dynamics that form the low-pass filter of Eq. 2.11, the gain magnitude, M, and phase angle, ϕ, are

$$M = |G(j\omega)| = \frac{a}{\sqrt{\omega^2 + a^2}}$$
$$\phi = \angle G(j\omega) = -\tan^{-1}\frac{\omega}{a}.$$

Any stable transfer function's long-term steady-state response to a sine wave input at frequency ω is a sine wave output at the same frequency, multiplied by the gain magnitude, M, and shifted by the phase angle, ϕ, as

$$\sin(\omega t) \xrightarrow{G} M \sin(\omega t + \phi), \tag{2.12}$$

in which the angle is given in radians. For example, if the phase lags by one-half of a cycle, $\phi = -\pi \equiv -180°$, then $M \sin(\omega t + \phi) = -M \sin(\omega t)$.

Chapter 3
Basic Control Architecture

3.1 Open-Loop Control

Suppose a system benefits by tracking relatively slow oscillatory environmental fluctuations at frequency ω_e and ignoring much faster noisy environmental fluctuations at frequency ω_n. Assume that the system has an intrinsic daily oscillator at frequency $\omega_0 = 1$, with time measured in days. How can a system build a control circuit that uses its intrinsic daily oscillator to track slower environmental signals and ignore faster noisy signals?

We can begin by considering circuit designs that follow the cascade in Fig. 2.1b. That cascade is a single direct path from input to output, matching the cascade in Eq. 2.3. That path is an open loop because there is no closed-loop feedback.

Using the components in Fig. 2.1b, the internal oscillator is given by

$$P(s) = \frac{\omega_0}{s^2 + \omega_0^2},$$

and the external reference signal is given by

$$R(s) = \frac{\omega_e}{s^2 + \omega_e^2} + \frac{\omega_n}{s^2 + \omega_n^2},$$

the sum of one low- and one high-frequency sine wave. From Fig. 2.1b, the design goal seeks to create a preprocess controlling filter, $C(s)$, that combines with the intrinsic internal oscillator, $P(s)$, to transform the reference input, $R(s)$, into an output, $Y(s) \approx \omega_e/(s^2 + \omega_e^2)$, that fluctuates at ω_e and ignores ω_n.

In this case, we know exactly the intrinsic dynamics, $P(s)$. Thus, we can use the open-loop path in Fig. 2.1b to find a controller, $C(s)$, such that the transfer function $C(s)P(s)$ gives approximately the input–output relation that we seek between $R(s)$ and $Y(s)$. For example, by using the controller

Fig. 3.1 Bode plot of an intrinsic oscillator, $P(s)$, modulated by a controller, $C(s)$, in an open loop $L(s) = C(s)P(s)$. The gold curves follow Eq. 3.3, in which the actual frequency of the internal oscillator is $\tilde{\omega}_0 = 1.2$ rather than the value $\omega_0 = 1$ that set the design of the controller. The underlying blue curves show the outcome when the internal oscillator frequency matches the design frequency, $\tilde{\omega}_0 = \omega_0 = 1$
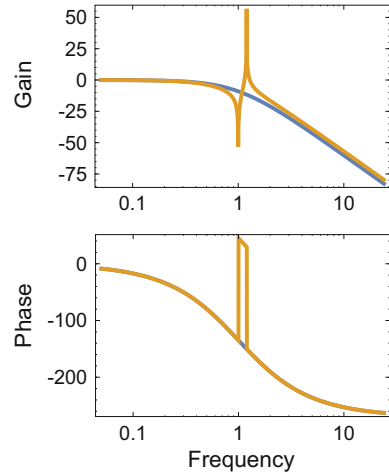
$$C(s) = \left(\frac{\omega_0}{s + \omega_0}\right)^3 \left(\frac{s^2 + \omega_0^2}{\omega_0}\right), \tag{3.1}$$

the open-loop system becomes

$$L(s) = C(s)P(s) = \left(\frac{\omega_0}{s + \omega_0}\right)^3, \tag{3.2}$$

because the second term in $C(s)$ cancels $P(s)$. The system $L(s)$ is the low-pass filter in Eq. 2.11 raised the third power. With $\omega_0 = 1$, this system has a Bode plot similar to the blue curve in Fig. 2.2e, f, but because of the exponent in $L(s)$, the gain falls more quickly at high frequencies and the phase lag is greater.

As with the low-pass filter illustrated in Fig. 2.2, this open-loop system, $L(s)$, tracks environmental signals at frequency $\omega_e \ll \omega_0$ and suppresses noisy signals at frequency $\omega_n \gg \omega_0$. However, even if we could create this controller over the required range of frequencies, it might turn out that this system is fragile to variations in the parameters.

We could study robustness by using the differential equations to calculate the dynamics for many combinations of parameters. However, such calculations are tedious, and the analysis can be difficult to evaluate for more than a couple of parameters. Using Bode plots provides a much easier way to analyze system response under various conditions.

Suppose, for example, that in the absence of inputs, the internal oscillator, $P(s)$, actually fluctuates at the frequency $\tilde{\omega}_0 \neq \omega_0$. Then, the open-loop system becomes

$$L(s) = \frac{\tilde{\omega}_0}{\omega_0} \left(\frac{\omega_0}{s + \omega_0} \right)^3 \left(\frac{s^2 + \omega_0^2}{\omega_0} \right) \left(\frac{\tilde{\omega}_0}{s^2 + \tilde{\omega}_0^2} \right), \tag{3.3}$$

in which the first term adjusts the gain to be one at $s = 0$.

The gold curves in Fig. 3.1 show the Bode plot for this open loop, using $\omega_0 = 1$ and $\tilde{\omega}_0 = 1.2$. Note the resonant peak in the upper magnitude plot. That peak occurs when the input frequency matches the natural frequency of the intrinsic oscillator, $\tilde{\omega}_0$. Near that resonant frequency, the system "blows up," because the denominator in the last term, $s^2 + \tilde{\omega}_0^2$, goes to zero as $s = j\omega \to j\tilde{\omega}_0$ and $s^2 \to -\tilde{\omega}_0^2$.

In summary, open-loop control works well when one has accurate information. Successful open-loop control is simple and has relatively low cost. However, small variations in the intrinsic process or the modulating controller can cause poor performance or instabilities, leading to system failure.

3.2 Feedback Control

Feedback and feedforward have different properties. Feedforward action is obtained by matching two transfer functions, requiring precise knowledge of the process dynamics, while feedback attempts to make the error small by dividing it by a large quantity.

—Åström and Murray (2008, p. 320)

Feedback often solves problems of uncertainty or noise. Human-designed systems and natural biological systems frequently use feedback control.

Figure 2.1c shows a common form of negative feedback. The output, y, is returned to the input. The output is then subtracted from the environmental reference signal, r. The new system input becomes the error between the reference signal and the output, $e = r - y$.

In closed-loop feedback, the system tracks its target reference signal by reducing the error. Any perturbations or uncertainties can often be corrected by system dynamics that tend to move the error toward zero. By contrast, a feedforward open loop has no opportunity for correction. Feedforward perturbations or uncertainties lead to uncorrected errors.

In the simple negative feedback of Fig. 2.1c, the key relation between the open-loop system, $L(s) = C(s)P(s)$, and the full closed-loop system, $G(s)$, is

$$G(s) = \frac{L(s)}{1 + L(s)}. \tag{3.4}$$

This relation can be derived from Fig. 2.1c by noting that, from the error input, $E(s)$, to the output, $Y(s)$, we have $Y = LE$ and that $E = R - Y$. Substituting the second equation into the first yields $Y = L(R - Y)$. Solving for the output Y relative to the input R, which is $G = Y/R$, yields Eq. 3.4.

The error, E, in response to the environmental reference input, R, can be obtained by a similar approach, yielding

$$E(s) = \frac{1}{1 + L(s)} R(s). \qquad (3.5)$$

If the open loop, $L(s)$, has a large gain, that gain will divide the error by a large number and cause the system to track closely to the reference signal. A large gain for $L = CP$ can be achieved by multiplying the controller, C, by a large constant, k. The large gain causes the system to respond rapidly to deviations from the reference signal.

Feedback, with its powerful error correction, typically provides good performance even when the actual system process, P, or controller, C, differs from the assumed dynamics. Feedback also tends to correct for various types of disturbances and noise, and can also stabilize an unstable open-loop system.

Feedback has two potential drawbacks. First, implementing feedback may require significant costs for the sensors to detect the output and for the processes that effectively subtract the output value from the reference signal. In electronics, the implementation may be relatively simple. In biology, feedback may require various additional molecules and biochemical reactions to implement sensors and the flow of information through the system. Simple open-loop feedforward systems may be more efficient for some problems.

Second, feedback can create instabilities. For example, when $L(s) \to -1$, the denominator of the closed-loop system in Eq. 3.4 approaches zero, and the system blows up. For a sinusoidal input, if there is a frequency, ω, at which the magnitude, $|L(j\omega)|$, is one and the phase is shifted by one-half of a cycle, $\phi = \pm\pi = \pm 180°$, then $L(j\omega) = -1$.

The problem of phase arises from the time lag (or lead) between input and feedback. When the sinusoidal input is at a peak value of one, the output is shifted to a sinusoidal trough value of minus one. The difference between input and output combines in an additive, expansionary way rather than providing an error signal that can shrink toward an accurate tracking process. In general, time delays in feedback can create instabilities.

Instabilities do not require an exact half cycle phase shift. Suppose, for example, that the open loop is

$$L(s) = \frac{k}{(s+1)^3}.$$

This system is stable, because its eigenvalues are the roots of the polynomial in the denominator, in this case $s = -1$, corresponding to a strongly stable system. The closed loop has the transfer function

$$G(s) = \frac{L(s)}{1 + L(s)} = \frac{k}{k + (s+1)^3},$$

which has an eigenvalue with real part greater than zero for $k > 8$, causing the system to be unstable. An unstable system tends to explode in magnitude, leading to system failure or death.

3.3 Proportional, Integral, and Derivative Control

Open loop systems cannot use information about the error difference between the target reference input and the actual output. Controllers must be designed based on information about the intrinsic process and the likely inputs.

By contrast, feedback provides information about errors, and controller design focuses primarily on using the error input. Given the error, the controller outputs a new command reference input to the intrinsic system process. Precise knowledge about the intrinsic system dynamics is much less important with feedback because the feedback loop can self-correct.

This section discusses controller design for feedback systems. A controller is a process that modulates system dynamics. For the simplest feedback shown in Fig. 2.1c, we start with an intrinsic process, $P(s)$, and end up with feedback system dynamics

$$G(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{L(s)}{1 + L(s)},$$

in which $C(s)$ is the controller. The problem is how to choose a process, $C(s)$, that balances the tradeoffs between various measures of success, such as tracking the reference input and robustness to perturbations and uncertainties.

Figure 3.2a includes two kinds of perturbations. The input d describes the load disturbance, representing uncertainties about the internal process, $P(s)$, and disturbances to that internal process. Traditionally, one thinks of d as a relatively low-frequency perturbation that alters the intrinsic process. The input n describes perturbations that add noise to the sensor that measures the process output, η, to yield the final output, y. That measured output, y, is used for feedback into the system.

To analyze alternative controller designs, it is useful to consider how different controllers alter the open-loop dynamics, $L(s) = C(s)P(s)$. How does a particular change in the controller, $C(s)$, modulate the intrinsic dynamics, $P(s)$?

First, we can simply increase the gain by letting $C(s) = k_p > 1$, a method called proportional control. The system becomes $G = k_p P/(1 + k_p P)$. For large k_p and positive $P(s)$, the system transfer function is $G(s) \to 1$, which means that the system output tracks very closely to the system input. Proportional control can greatly improve tracking at all frequencies. However, best performance often requires tracking low-frequency environmental inputs and ignoring noisy high-frequency inputs from the reference signal. In addition, large k_p values can cause instabilities, and it may be that $P(s) < 0$ for some inputs.
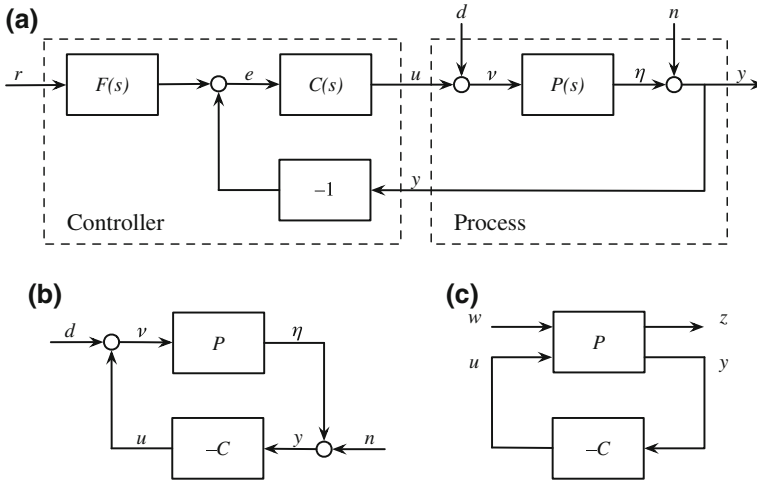
Fig. 3.2 Closed-loop feedback. **a** An extended feedback loop with inputs for disturbance, d, and noise, n. The function $F(s)$ may be used to filter the reference input, providing a second degree of freedom in addition to the main controller, $C(s)$. The system can be divided into intrinsic processes that cannot be adjusted directly and designed processes of control that can be adjusted. Note the inputs for each block: r and y for the controller, and u, d, and n for the process. **b** In this panel, the blocks P and C represent the multicomponent process and control blocks from the upper panel. The reference signal is assumed to be zero, allowing one to focus on the roles of disturbance and noise in relation to system stability. **c** An abstraction of the feedback process, in which the vector y includes all the signals from the process to the controller, u includes all the control input signals to the process, w includes all the extrinsic inputs, and z includes any additional signal outputs from the process. Redrawn from Åström and Murray (2008), © Princeton University Press

Second, we can add integral control by including the term k_i/s to the controller. We can understand why this term is an integrator by considering a few steps of analysis that extend earlier equations. Multiplying Eq. 2.5 by $1/s$ increases the denominator's order of its polynomial in s. That increase in the exponents of s corresponds to an increase in the order of differentiation for each term on the left side of Eq. 2.4, which is equivalent to integrating each term on the right side of that equation. For example, if we start with $\dot{x} = u$ and then increase the order of differentiation on the left side, $\ddot{x} = u$, this new expression corresponds to the original expression with integration of the input signal, $\dot{x} = \int u \, dt$.

Integrating the input smooths out high-frequency fluctuations, acting as a filter that passes low-frequency inputs and blocks high-frequency inputs. Integration causes a slower, smoother, and often more accurate adjustment to the input signal. A term such as $a/(s + a)$ is an integrator for large s and a pass-through transfer function with value approaching one for small s.

Perfect tracking of a constant reference signal requires a pure integrator term, $1/s$. A constant signal has zero frequency, $s = 0$. To track a signal perfectly, the system transfer function's gain must be one so that the output equals the input. For the simple closed loop in Eq. 3.4, at zero frequency, $G(0)$ must be one. The tracking error is $1 - G = 1/(1 + L)$. The error goes to zero as the gain of the open loop goes to infinity, $L(0) \to \infty$. A transfer function requires a term $1/s$ to approach infinity as s goes to zero. In general, high open loop gain leads to low tracking error.

Third, we can add derivative control by including the term $k_d s$. We can understand why this term differentiates the input term by following the same steps as for the analysis of integration. Multiplying Eq. 2.5 by s increases the numerator's order of its polynomial in s. That increase in the exponents of s corresponds to an increase in the order of differentiation for each term on the right side of Eq. 2.4. Thus, the original input term, $u(t)$, becomes the derivative with respect to time, $\dot{u}(t)$.

Differentiating the input causes the system to respond to the current rate of change in the input. Thus, the system responds to a prediction of the future input, based on a linear extrapolation of the recent trend.

This leading, predictive response enhances sensitivity to short-term, high-frequency fluctuations and tends to block slow, low-frequency input signals. Thus, differentiation acts as a high-pass filter of the input signal. A term such as $s + a$ multiplies signals by a for low-frequency inputs and multiplies signals by the increasing value of $s + a$ for increasingly high-frequency inputs. Differentiators make systems very responsive, but also enhance sensitivity to noisy high-frequency perturbations and increase the tendency for instability.

A basic proportional, integral, and derivative (PID) controller has the form

$$C(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s}. \tag{3.6}$$

PID controllers are widely used across all engineering applications. They work reasonably well for many cases, they are relatively easy to understand, and their parameters are relatively easy to tune for various tradeoffs in performance.

3.4 Sensitivities and Design Tradeoffs

Figure 3.2a shows a basic feedback loop with three inputs: the reference signal, r, the load disturbance, d, and the sensor noise, n. How do these different signals influence the error between the reference signal and the system output? In other words, how sensitive is the system to these various inputs?

To derive the sensitivities, define the error in Fig. 3.2a as $r - \eta$, the difference between the reference input, r, and the process output, η (Åström and Murray 2008, Sect. 11.1). To obtain the transfer function between each input and output, we use the rule for negative feedback: The transfer function between the input and output is the

open loop directly from the input to the output, L, divided by one plus the pathway around the feedback loop, $1 + L$.

If we assume in Fig. 3.2a that there is no feedforward filter, so that $F = 1$, and we define the main open loop as $L = CP$, then the output η in response to the three inputs is

$$\eta = \frac{L}{1 + L}r + \frac{P}{1 + L}d - \frac{L}{1 + L}n, \tag{3.7}$$

in which each term is the open loop between the input signal and the output, η, divided by one plus the pathway around the full loop, L. If we define

$$S = \frac{1}{1 + L} \qquad T = \frac{L}{1 + L} \qquad S + T = 1, \tag{3.8}$$

with S as the sensitivity function and T as the complementary sensitivity function, then the error is

$$r - \eta = Sr - PSd + Tn. \tag{3.9}$$

This expression highlights the fundamental design tradeoffs in control that arise because $S + T = 1$. If we reduce T and the sensitivity to noise, we increase S. An increase in S raises the error in relation to the reference signal, r, and the error in relation to the load disturbance, d. If we reduce S, we increase T and the sensitivity to noise, n. These sensitivity tradeoffs suggest two approaches to design.

First, the sensitivities $S(s)$ and $T(s)$ depend on the input, s. Thus, we may adjust the tradeoff at different inputs. For example, we may consider inputs, $s = j\omega$, at various frequencies, ω. Sensor noise, n, often arises as a high frequency disturbance, whereas the reference input, r, and the load disturbance, d, often follow a low frequency signal. If so, then we can adjust the sensitivity tradeoff to match the common input frequencies of the signals. In particular, at low frequency for which r and d dominate, we may choose low S values whereas, at high frequency for which n dominates, we may choose low T values.

Second, we may add an additional control process that alters the sensitivity tradeoff. For example, we may use the feedforward filter, F, in Fig. 3.2a, to modulate the reference input signal. With that filter, the transfer function from the input, r, to the error output, $r - \eta$, becomes $1 - FT$. If we know the form of T with sufficient precision, we can choose $FT \approx 1$, and thus we can remove the sensitivity of the error to the reference input.

Note that adjusting the tradeoff between S and T only requires an adjustment to the loop gain, L, which usually does not require precise knowledge about the system processes. By contrast, choosing F to cancel the reference input requires precise information about the form of T and the associated system processes. In other words, feedback is relatively easy and robust because it depends primarily on adjusting gain magnitude, whereas feedforward requires precise knowledge and is not robust to misinformation or perturbation.

Chapter 4
PID Design Example

I illustrate the principles of feedback control with an example. We start with an intrinsic process

$$P(s) = \left(\frac{a}{s+a}\right)\left(\frac{b}{s+b}\right) = \frac{ab}{(s+a)(s+b)}.$$

This process cascades two exponential decay systems, each with dynamics as in Eq. 2.8 and associated transfer function as in Eq. 2.9. For example, if the input into this system is a unit impulse at time zero, then the system output is

$$y(t) = \frac{ab}{b-a}\left(e^{-at} - e^{-bt}\right),$$

expressing the cascade of two exponentially decaying processes.

For this example, we use

$$P(s) = \frac{1}{(s+0.1)(s+10)} \tag{4.1}$$

as the process. We also consider an alternative process

$$\tilde{P}(s) = \frac{1}{(s+0.01)(s+100)}. \tag{4.2}$$

We assume during system analysis and design that Eq. 4.1 describes the process, but in fact Eq. 4.2 is actually the true process. Put another way, the difference between the two processes may reflect uncertain information about the true process or unknown disturbances that alter the process. Thus, we may consider how a system performs
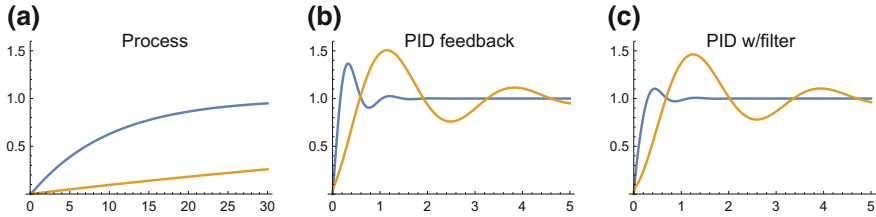
Fig. 4.1 Response of the system output, $\eta = y$, to a sudden unit step increase in the reference input, r, in the absence of disturbance and noise inputs, d and n. The x-axis shows the time, and the y-axis shows the system output. **a** Response of the original process, $P(s)$, in Eq. 4.1 (blue curve) and of the process with altered parameters, $\tilde{P}(s)$ in Eq. 4.2 (gold curve). **b** System with the PID controller embedded in a negative feedback loop, with no feedforward filter, $F(s) = 1$, as in Fig. 3.2a. **c** PID feedback loop with feedforward filter, F, in Eq. 4.4

when it was designed, or evolved, in response to a process, P, and the underlying system becomes $\tilde{P}$.

In this example, the problem concerns the design of a negative feedback loop, as in Fig. 3.2a, that uses a controller with proportional, integral, and derivative (PID) action. Many methods derive PID controllers by tuning the various sensitivity and performance tradeoffs (Åström and Hägglund 2006; Garpinger et al. 2014).

I obtained the parameters for the PID controller in Eq. 3.6 by using the Ziegler–Nichols method in Mathematica, yielding

$$C(s) = \frac{6s^2 + 121s + 606}{s}. \qquad (4.3)$$

I also used Mathematica to calculate the feedforward filter in Fig. 3.2a, yielding

$$F(s) = \frac{s^2 + 10.4s + 101}{s^2 + 20.2s + 101}. \qquad (4.4)$$

4.1   Output Response to Step Input

Figure 4.1 illustrates various system responses to a unit step increase from zero to one in the reference input signal, r. Panel (a) shows the response of the base process, P, by itself. The blue curve is the double exponential decay process of Eq. 4.1. That process responds slowly because of the first exponential process with time decay $a = 0.1$, which averages inputs over a time horizon with decay time $1/a = 10$, as in Eq. 2.8.

The gold curve, based on Eq. 4.2, rises even more slowly, because that alternative process, $\tilde{P}$, has an even longer time horizon for averaging inputs of $1/a = 100$.

Panel (b) shows the response of the full feedback loop of Fig. 3.2a with the PID controller in Eq. 4.3 and no feedforward filter, $F = 1$. Note that the system responds much more rapidly, with a much shorter time span over the x-axis than in (a). The rapid response follows from the very high gain of the PID controller, which strongly amplifies low-frequency inputs.

The PID controller was designed to match the base process P in Eq. 4.1, with response in blue. When the actual base process deviates as in $\tilde{P}$ of Eq. 4.2, the response is still reasonably good, although the system has a greater overshoot upon first response and takes longer to settle down and match the reference input. The reasonably good response in the gold curve shows the robustness of the PID feedback loop to variations in the underlying process.

Panel (c) shows the response of the system with a feedforward filter, F, from Eq. 4.4. Note that the system in blue with the base process, P, improves significantly, with lower overshoot and less oscillation when settling to match the reference input. By contrast, the system in gold with the alternative base process, $\tilde{P}$, changes its response very little with the additional feedforward filter. This difference reflects the fact that feedforward works well only when one has very good knowledge of the underlying process, whereas feedback works broadly and robustly with respect to many kinds of perturbations.

4.2   Error Response to Noise and Disturbance

Figure 4.2 illustrates the system error in response to sensor noise, n, and process disturbance, d. Panel (a) shows the error in response to a unit step change in n, the input noise to the sensor. That step input to the sensor creates a biased measurement, y, of the system output, η. The biased measured value of y is fed back into the control loop. A biased sensor produces an error response that is equivalent to the output response for a reference signal. Thus, Fig. 4.2a matches Fig. 4.1b.

Panel (b) shows the error response to an impulse input at the sensor. An impulse causes a brief jolt to the system. The system briefly responds by a large deviation from its setpoint, but then returns quickly to stable zero error, at which the output matches the reference input. An impulse to the reference signal produces an equivalent deviation in the system output but with opposite sign.

The error response to process disturbance in panels (c) and (d) demonstrates that the system strongly rejects disturbances or uncertainties to the intrinsic system process.
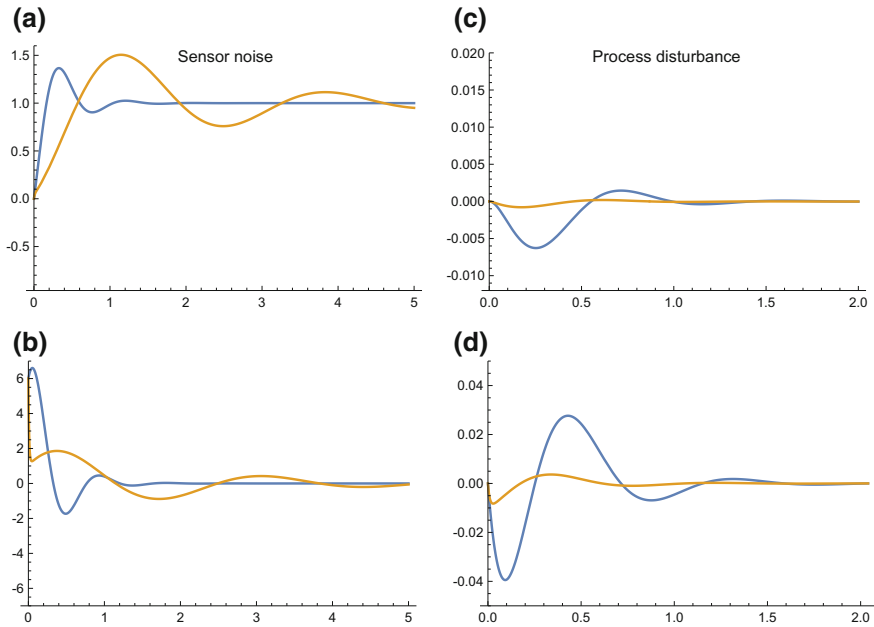
Fig. 4.2 Error response, $r - \eta$, of the PID feedback loop to sensor noise, n, or process disturbance, d, from Eq. 3.9. Blue curve for the process, P, in Eq. 4.1 and gold curve for the altered process, $\tilde{P}$, in Eq. 4.2. **a** Error response to sensor noise input, n, for a unit step input and **b** for an impulse input. **c** Error response to process disturbance input, d, for a unit step input and **d** for an impulse input. An impulse is $u(t)\mathrm{d}t = 1$ at $t = 0$ and $u(t) = 0$ at all other times. The system responses in gold curves reflect the slower dynamics of the altered process. If the altered process had faster intrinsic dynamics, then the altered process would likely be more sensitive to noise and disturbance
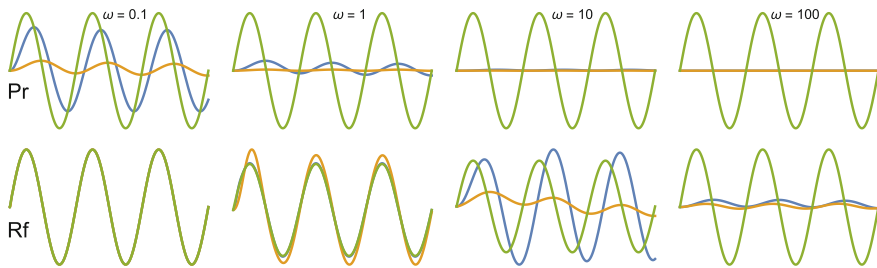
Fig. 4.3 System response output, $\eta = y$, to sine wave reference signal inputs, r. Each column shows a different frequency, ω. The rows are (Pr) for reference inputs into the original process, P or $\tilde{P}$, without a modifying controller or feedback loop, and (Rf) for reference inputs into the closed-loop feedback system with the PID controller in Eq. 4.3. The green curve shows the sine wave input. The blue curve shows systems with the base process, P, from Eq. 4.1. The gold curve shows systems with the altered process, $\tilde{P}$, from Eq. 4.2. In the lower left panel, all curves overlap. In the lower panel at $\omega = 1$, the green and blue curves overlap. In the two upper right panels, the blue and gold curves overlap near zero

4.3 Output Response to Fluctuating Input

Figure 4.3 illustrates the system output in response to fluctuating input (green). The top row shows the output of the system process, either P (blue) or $\tilde{P}$ (gold), alone in an open loop. The system process is a cascade of two low-pass filters, which pass low-frequency inputs and do not respond to high-frequency inputs.

The upper left panel shows the response to the (green) low-frequency input, $\omega = 0.1$, in which the base system P (blue) passes through the input with a slight reduction in amplitude and lag in phase. The altered system $\tilde{P}$ (gold) responds only weakly to the low frequency of $\omega = 0.1$, because the altered system has slower response characteristics than the base system. At a reduced input frequency of $\omega = 0.01$ (not shown), the gold curve would match the blue curve at $\omega = 0.1$. As frequency increases along the top row, the processes P and $\tilde{P}$ block the higher-frequency inputs.

The lower row shows the response of the full PID feedback loop system. At a low frequency of $\omega \leq 0.1$, the output tracks the input nearly perfectly. That close tracking arises because of the very high gain amplification of the PID controller at low frequency, which reduces the system tracking error to zero, as in Eq. 3.5.

At a higher frequency of $\omega = 10$, the system with the base process P responds with a resonant increase in amplitude and a lag in phase. The slower altered process, $\tilde{P}$, responds only weakly to input at this frequency. As frequency continues to increase, both systems respond weakly or not at all.

The system response to sensor noise would be of equal magnitude but altered sign and phase, as shown in Eq. 3.7.

Low-frequency tracking and high-frequency rejection typically provide the greatest performance benefit. The environmental references that it pays to track often change relatively slowly, whereas the noisy inputs in both the reference signal and in the sensors often fluctuate relatively rapidly.

4.4 Insights from Bode Gain and Phase Plots

Figure 4.4 provides more general insight into the ways in which PID control, feedback, and input filtering alter system response.

Panels (a) and (b) show the Bode gain and phase responses for the intrinsic system process, P (blue), and the altered process, $\tilde{P}$ (gold). Low-frequency inputs pass through. High-frequency inputs cause little response. The phase plot shows that these processes respond slowly, lagging the input. The lag increases with frequency.

Panels (c) and (d) show the responses for the open loop with the PID controller, C, combined with the process, P or $\tilde{P}$, as in Fig. 2.1b. Note the very high gain in panel (c) at lower frequencies and the low gain at high frequencies.
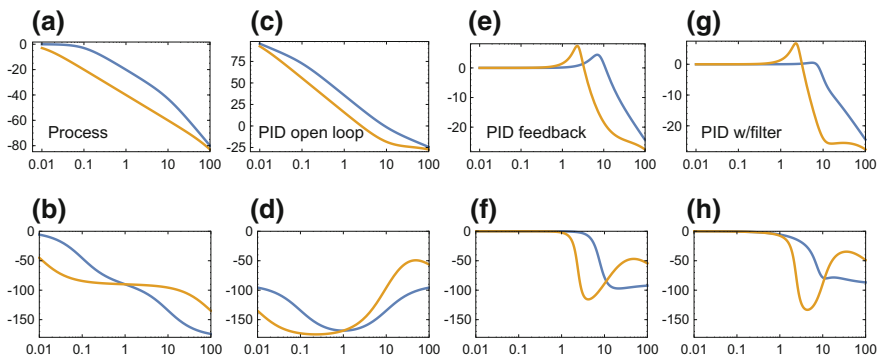
Fig. 4.4 Bode gain (top) and phase (bottom) plots for system output, $\eta = y$, in response to reference input, r, in the absence of load disturbance and sensor noise. Blue curves for systems with the base process, P, in Eq. 4.1. Gold curves for systems with the altered process, $\tilde{P}$, in Eq. 4.2. **a**, **b** The original unmodified process, P or $\tilde{P}$, with no controller or feedback. **c**, **d** The open loop with no feedback, CP or $C\tilde{P}$, with the PID controller, C, in Eq. 4.3. **e**, **f** The closed loop with no feedforward filter, $F = 1$. **g**, **h** The closed loop with the feedforward filter, F, in Eq. 4.4

PID controllers are typically designed to be used in closed-loop feedback systems, as in Fig. 2.1c. Panels (e) and (f) illustrate the closed-loop response. The high open-loop gain of the PID controller at low frequency causes the feedback system to track the reference input closely. That close tracking matches the $\log(1) = 0$ gain at low frequency in panel (e). Note also the low-frequency phase matching, or zero phase lag, shown in panel (f), further demonstrating the close tracking of reference inputs. At high frequency, the low gain of the open-loop PID controller shown in panel (c) results in the closed-loop rejection of high-frequency inputs, shown as the low gain at high frequency in panel (e).

Note the resonant peak of the closed-loop system in panel (e) near $\omega = 10$ for the blue curve and at a lower frequency for the altered process in the gold curve. Note also that the altered process, $\tilde{P}$, in gold, retains the excellent low-frequency tracking and high-frequency input rejection, even though the controller was designed for the base process, P, shown in blue. The PID feedback loop is robust to differences in the underlying process that varies from the assumed form of P.

Panels (g) and (h) show the PID closed-loop system with a feedforward filter, F, as in Fig. 3.2a. The feedforward filter smooths out the resonant peak for the blue curve, so that system does not amplify inputs at resonant frequencies. Amplified resonant inputs may lead to instabilities or poor system performance. Note that the feedforward filter does not have much effect on the altered process in gold. Feedforward modifiers of a process typically work well only for a specific process. They often do not work robustly over a variant range of processes.

4.5   Sensitivities in Bode Gain Plots

Figure 4.5 illustrates the sensitivities of the system error output, $r - \eta$, to inputs from the reference, r, sensor noise, n, and load disturbance, d, signals, calculated from Eq. 3.9. Figure 3.2a shows the inputs and loop structure.

The blue curve of panel (a) shows the error sensitivity to the reference input. That sensitivity is approximately the mirror image of the system output response to the reference input, as shown in Fig. 4.4e (note the different scale). The duality of the error response and the system response arises from the fact that the error is $r - \eta$, and the system response is η.

Perfect tracking means that the output matches the input, $r = \eta$. Thus, a small error corresponds to a low gain of the error in response to input, as occurs at low frequency for the blue curve of Fig. 4.5a. In the same way, a small error corresponds to a gain of one for the relation between the reference input, r, and the system output, η, as occurs at low frequency for the blue curve of Fig. 4.4e.

The noise sensitivity in the green curve of Fig. 4.5a shows that the system error is sensitive to low-frequency bias in the sensor measurements, y, of the system output, η. When the sensor produces a low-frequency bias, that bias feeds back into the system and creates a bias in the error estimate, thus causing an error mismatch between the reference input and the system output. In other words, the system is sensitive to errors when the sensor suffers low-frequency perturbations. The PID system rejects high-frequency sensor noise, leading to the reduced gain at high frequency illustrated by the green curve.

The disturbance load sensitivity in the red curve of Fig. 4.5a shows the low sensitivity of this PID feedback system to process variations.

This PID feedback system is very robust to an altered underlying process, as shown in earlier figures. Here, Fig. 4.5b illustrates that robustness by showing the relatively minor changes in system sensitivities when the underlying process changes
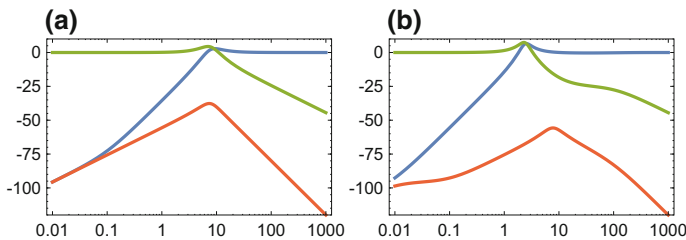
Fig. 4.5 Bode gain plots for the error output, $r - \eta$, in response to reference input, r (blue), sensor noise, n (green), and load disturbance, d (red), from Eq. 3.9. The systems are the full PID-controlled feedback loops as in Fig. 3.2a, with no feedforward filter. The PID controller is given in Eq. 4.3. **a** System with the base process, P, from Eq. 4.1. **b** System with the altered process, $\tilde{P}$, from Eq. 4.2

from P to $\tilde{P}$. However, other types of change to the underlying process may cause greater changes in system performance. Robustness depends on both the amount of change and the kinds of change to a system.

Chapter 5
Performance and Robustness Measures

A theory of design tradeoffs requires broadly applicable measures of cost, performance, stability, and robustness. For example, the PID controller in the previous example performs reasonably well, but we ignored costs. That PID controller achieved good tracking performance by using high gain amplification of low-frequency input signals. High gain in a negative feedback loop quickly drives the error to zero.

High gain has two potential problems. First, high signal amplification may require excessive energy in physical or biological systems. We must consider those costs for a high gain controller.

Second, high gain can cause system instability, with potential for system failure. We must consider the tradeoff between the benefits of high gain and the loss of robustness against perturbations or uncertainties in system dynamics.

Beyond the simple PID example, we must consider a variety of tradeoffs in performance and robustness (Zhou and Doyle 1998; Qiu and Zhou 2010). Earlier, I discussed tradeoffs in system sensitivities to disturbance and noise. I also presented qualitative descriptions of system performance in terms of response time and tracking performance.

To advance the theory, we need specific measures of cost, performance, stability and robustness. We also need techniques to find optimal designs in relation to those conflicting measures of system attributes.

We will never find a perfect universal approach. There are too many dimensions of costs and benefits, and too many alternative ways to measure system attributes. Nonetheless, basic measures and simple optimization methods provide considerable insight into the nature of design. Those insights apply both to the building of human-designed systems to achieve engineering goals and to the interpretation and understanding of naturally designed biological systems built by evolutionary processes.

5.1 Performance and Cost: $\mathcal{J}$

To analyze performance, we must measure the costs and benefits associated with a particular system. We often measure those costs and benefits by the distance between a system's trajectory and some idealized trajectory with zero cost and perfect performance.

Squared deviations provide a distance measure between the actual trajectory and the idealized trajectory. Consider, for example, the control signal, $u(t)$, which the controller produces to feed into the system process, as in Fig. 2.1c.

The value of $|u(t)|^2 = u^2$ measures the magnitude of the signal as a squared distance from zero. We can think of u^2 as the instantaneous power of the control signal. Typically, the power requirements for control are a cost to be minimized.

The square of the error output signal, $|e(t)|^2 = e^2$, measures the distance of the system from the ideal performance of $e = 0$. Minimizing the squared error maximizes performance. Thus, we may think of performance at any particular instant, t, in terms of the cost function

$$\mathcal{J}(t) = u^2 + \rho^2 e^2,$$

for which minimum cost corresponds to maximum performance. Here, ρ is a weighting factor that determines the relative value of minimizing the control signal power, u^2, versus minimizing the tracking error, e^2.

Typically, we measure the cost function over a time interval. Summing up $\mathcal{J}(t)$ continuously from $t = 0$ to T yields

$$\mathcal{J} = \int_0^T (u^2 + \rho^2 e^2)\mathrm{d}t. \tag{5.1}$$

Most squared distance or quadratic performance analyses arise from extensions of this basic equation. Given this measure, optimal design trades off minimizing the energy cost to drive the system versus maximizing the benefit of tracking a target goal.

5.2 Performance Metrics: Energy and $\mathcal{H}_2$

The cost measure in Eq. 5.1 analyzes signals with respect to time. It is natural to think of inputs and outputs as changing over time. With temporal dynamics, we can easily incorporate multivariate signals and nonlinearities. In spite of those advantages, we often obtain greater insight by switching to a frequency analysis of signals, as in the previous chapters.

In this section, I present alternative measures of cost and performance in terms of transfer functions and complex signals. Those alternative measures emphasize frequencies of fluctuations rather than changes through time. Frequency and complex

analysis allow us to take advantage of transfer functions, Bode plots, and other powerful analytical tools that arise when we assume linear dynamics.

The assumption of linearity does not mean that we think the actual dynamics of physical and biological processes are linear. Instead, starting with the linear case provides a powerful way in which to gain insight about dynamics.

In the previous section, we considered how to measure the magnitude of fluctuating control and error signals. A magnitude that summarizes some key measure is often called a *norm*. In the prior section, we chose the sum of squared deviations from zero, which is related to the 2–norm of a signal

$$\|u(t)\|_2 = \left(\int_0^\infty |u(t)|^2 \mathrm{d}t \right)^{1/2}. \tag{5.2}$$

The energy of the signal is the square of the 2–norm, $\|u(t)\|_2^2$. When the time period in the cost function of Eq. 5.1 goes to infinity, $T \to \infty$, we can write the cost function as

$$\mathcal{J} = \|u(t)\|_2^2 + \rho^2 \|e(t)\|_2^2. \tag{5.3}$$

The signal $u(t)$ is a function of time. The associated transfer function $U(s)$ describes exactly the same signal, but as a function of the complex number, s, rather than of time, t.

It is often much easier to work with the transfer function for analysis, noting that we can go back and forth between time and transfer function descriptions. For the analysis of squared distance metrics, the 2–norm of the transfer function expression is

$$\|U(s)\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^\infty |U(j\omega)|^2 \mathrm{d}\omega \right)^{1/2}. \tag{5.4}$$

This transfer function 2–norm is often referred to as the $\mathcal{H}_2$ norm. The term $|U(j\omega)|^2$ is the square of the Bode gain or magnitude, as in Fig. 2.2e. That gain describes the amplification of a sinusoidal input at frequency ω. The $\mathcal{H}_2$ norm expresses the average amplification of input signals over all input frequencies.

If the goal is to minimize the control input signal, u, or the error deviation from zero, e, then the greater the amplification of a signal, the greater the cost. Thus, we can use the $\mathcal{H}_2$ norm to define an alternative cost function as

$$\mathcal{J} = \|U(s)\|_2^2 + \rho^2 \|E(s)\|_2^2, \tag{5.5}$$

which leads to methods that are often called $\mathcal{H}_2$ analysis. This cost describes the amplification of input signals with respect to control and error outputs when averaged overall input frequencies. Minimizing this cost reduces the average amplification of input signals.

If the energy 2–norm in Eq. 5.2 is finite, then the energy 2–norm and the $\mathcal{H}_2$ norm are equivalent, $\|u(t)\|_2 = \|U(s)\|_2$, and we can use Eqs. 5.3 and 5.5 interchangeably. Often, it is more convenient to work with the transfer function form of the $\mathcal{H}_2$ norm.

We can use any combination of signals in the cost functions. And we can use different weightings for the relative importance of various signals. Thus, the cost functions provide a method to analyze a variety of tradeoffs.

5.3 Technical Aspects of Energy and $\mathcal{H}_2$ Norms

I have given three different cost functions. The first in Eq 5.1 analyzes temporal changes in signals, such as $u(t)$, over a finite time interval. That cost function is the most general, in the sense that we can apply it to any finite signals. We do not require assumptions about linearity or other special attributes of the processes that create the signals.

The second function in Eq. 5.3 measures cost over an infinite time interval and is otherwise identical to the first measure. Why consider the unrealistic case of infinite time?

Often, analysis focuses on a perturbation that moves a stable system away from its equilibrium state. As the system returns to equilibrium, the error and control signals go to zero. Thus, the signals have positive magnitude only over a finite time period, and the signal energy remains finite. As noted above, if the energy 2–norm is finite, then the energy 2–norm and the $\mathcal{H}_2$ norm are equivalent, and the third cost function in Eq. 5.5 is equivalent to the second cost function in Eq. 5.3.

If the signal energy of the second cost function in Eq. 5.3 is infinite, then that cost function is not useful. In an unstable system, the error often grows with time, leading to infinite energy of the error signal. For example, the transfer function $1/(s-1)$ has temporal dynamics given by $y(t) = y(0)e^t$, growing exponentially with time. The system continuously amplifies an input signal, creating instability and an output signal with infinite energy.

When the energy is infinite, the $\mathcal{H}_2$ norm may remain finite. For the transfer function $1/(s-1)$, the $\mathcal{H}_2$ norm is $1/\sqrt{2}$. The average amplification of signals remains finite. In general, for a transfer function, $G(s)$, the $\mathcal{H}_2$ norm remains finite as long as $G(j\omega)$ does not go to infinity for any value of ω, and $G(j\omega) \to 0$ as $\omega \to \pm\infty$. Thus, the $\mathcal{H}_2$ norm cost in Eq. 5.5 can be used in a wider range of applications.

The $\mathcal{H}_2$ norm is related to many common aspects of signal processing and time series analysis, such as Fourier analysis, spectral density, and autocorrelation.

5.4 Robustness and Stability: $\mathcal{H}_\infty$

A transfer function for a system, $G(s)$, defines the system's amplification of input signals. For a sinusoidal input at frequency ω, the amplification, or gain, is the absolute value of the transfer function at that frequency, $|G(j\omega)|$.

Often, the smaller a system's amplification of inputs, the more robust the system is against perturbations. Thus, one common optimization method for designing controllers seeks to minimize a system's greatest amplification of inputs. Minimizing the greatest amplification guarantees a certain level of protection against the worst case perturbation. In some situations, one can also guarantee that a system is stable if its maximum signal amplification is held below a key threshold.

A system's maximum amplification of sinusoidal inputs over all input frequencies, ω, is called its $\mathcal{H}_\infty$ norm. For a system $G(s)$, the $\mathcal{H}_\infty$ norm is written as $\|G(s)\|_\infty$. The norm describes the maximum of $|G(j\omega)|$ over all ω. The maximum is also the peak gain on a Bode magnitude plot, which is equivalent to the resonance peak.

System stability and protection against perturbations set two fundamental criteria for system design. Thus, $\mathcal{H}_\infty$ methods are widely used in the engineering design of controllers and system architectures (Zhou and Doyle 1998).

Part II
Design Tradeoffs

Many performance tradeoffs occur. A system that responds quickly to control signals often suffers from sensitivity to perturbations. A more rapid response also associates with a greater tendency toward instability.

Design of a control system by an engineer must balance the competing dimensions of performance. Similarly, design of biological systems by evolutionary processes implicitly balances the different dimensions of success. In engineering, we can specify performance criteria. In biology, we must figure out how natural processes set the relative importance of different performance measures.

Once we have a set of performance criteria, how do we find the control architectures and parameters that perform well? If we do not have formal design methods, then we end up with ad hoc solutions. Such solutions may perform well. But we do not have any way to know if there are better solutions or better ways to formulate the design criteria.

Ideally, we would have an optimization method that provided the best solution for a given problem and a given set of performance criteria. Optimization forces us to specify the problem with clarity. We must write down exactly the performance criteria, the nature of the problem, and all associated assumptions. We then get an answer about whether there is a best design for the given assumptions or a set of comparable alternative designs.

Optimization is, of course, only as good as the assumptions that we make. In engineering, we may be able to specify design criteria clearly. Or, at least, we can experiment with various criteria and examine the alternative optimal designs.

In biology, figuring out the appropriate assumptions and constraints that express natural evolutionary processes can be very difficult. We may make some progress by trying different assumptions as hypotheses about the natural design process. We can then test the match between the optimal solutions and what we actually see in nature (Parker and Maynard Smith 1990).

Design by optimization must begin with performance criteria. Three kinds of performance criteria dominate in typical engineering applications.

Regulation, or homeostasis, concerns aspects of design that return a system to its setpoint. Good regulation requires insensitivity to perturbations. If the system does get pushed away from its setpoint, a well regulated system rapidly returns to its equilibrium. Tradeoffs arise between the responses to different kinds of perturbations.

Stabilization concerns aspects of design that protect against instability. An unstable system may lead to failure or death. Often, the primary design goal is to protect against instability.

Tracking concerns how well the system follows changes in environmental or reference input signals. A system that rapidly adjusts to changes may track closely to reference inputs but may suffer from sensitivity to perturbations or instability.

The next sections briefly illustrate these concepts. I use modified examples from the excellent article by Qiu and Zhou (2013).

Chapter 6
Regulation

The regulation problem analyzes how quickly a perturbed system returns to its equilibrium setpoint. For this problem, we assume that the setpoint does not change. We can, without loss of generality, assume that the external reference signal is $r = 0$.

With no external reference signal, we can express the general form of the regulation problem as in Fig. 6.1. We take the process, P, as given, subject to uncertainties or disturbances represented by the input, d. We seek an optimal controller, C, with respect to particular design tradeoffs.

6.1 Cost Function

The cost function summarizes the design tradeoffs. We use a cost function based on the $\mathcal{H}_2$ norm, similar to Eq. 5.5. The $\mathcal{H}_2$ norm describes the response of the system to perturbations when averaged over all input frequencies. Minimizing the $\mathcal{H}_2$ norm minimizes the extent to which the system responds to perturbations. Recall that the $\mathcal{H}_2$ norm is often equivalent to the signal energy, which is the total squared deviation of a signal from zero when measured from the time of an initial perturbation until the time when the signal returns to zero.

From Fig. 6.1, the two inputs are the load disturbance, d, and the sensor noise, n. The two outputs are the process output, η, and the control signal, u. We can write the outputs as transfer functions, $\eta(s)$ and $U(s)$, and the cost function in Eq. 5.5 as

$$\mathcal{J} = ||U(s)||_2^2 + \rho^2 ||\eta(s)||_2^2.$$

In this case, we need to relate each of the two outputs to each of the two inputs. We require four transfer functions to describe all of the input–output connections. For the transfer function between the input d and the output η, we write $G_{\eta d}(s)$, for
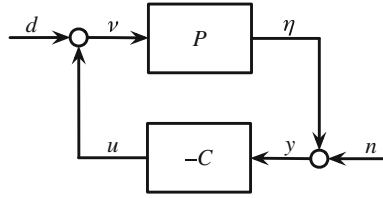
Fig. 6.1 Classic regulation problem illustrated by closed-loop feedback with a constant reference input signal, $r = 0$. The disturbance input, d, perturbs the system process. Such perturbations can be considered as stochasticity in the process, or as uncertainty with regard to the true process dynamics relative to the assumed dynamics. The noise input, n, perturbs the sensor that produces the output measurement, y, based on the actual process output, η. See Fig. 3.2 for context

which we assume that the other input, n, is zero. Using our usual rule for the transfer functions of a closed loop, the four functions are

$$G_{ud} = \frac{-PC}{1 + PC} \qquad\qquad G_{\eta d} = \frac{P}{1 + PC}$$

$$G_{un} = \frac{-C}{1 + PC} \qquad\qquad G_{\eta n} = \frac{-PC}{1 + PC}. \qquad (6.1)$$

We can express these transfer functions in terms of the sensitivities in Eq. 3.8 by defining the open loop as $L = PC$, the sensitivity as $S = 1/(1 + L)$, and the complementary sensitivity as $T = L/(1 + L)$, yielding

$$G_{ud} = -T \qquad\qquad G_{\eta d} = PS$$

$$G_{un} = -CS \qquad\qquad G_{\eta n} = -T. \qquad (6.2)$$

Because $S + T = 1$ at any input, s, these transfer functions highlight the intrinsic design tradeoffs.

We can now consider the total cost as the sum of the response with respect to the input d, holding n at zero, plus the response with respect to the input n, holding d at zero

$$\mathcal{J} = ||G_{ud}(s)||_2^2 + \rho^2 ||G_{\eta d}(s)||_2^2$$

$$+ ||G_{un}(s)||_2^2 + \rho^2 ||G_{\eta n}(s)||_2^2. \qquad (6.3)$$

For this example, we use impulse function inputs, $\delta(t)$, which provide a strong instantaneous shock to the system, as defined in the caption of Fig. 4.2. We can design the system to be relatively more or less sensitive to disturbance inputs relative to noise inputs by weighting the disturbance input by μ, so that $d(t) = \mu\delta(t)$ and $n(t) = \delta(t)$. Larger μ causes design by optimization to yield better disturbance regulation at the expense of worse noise regulation.

The transfer function for an impulse is equal to one. Thus, the transfer functions for disturbance and noise inputs are, respectively, $D(s) = \mu$ and $N(s) = 1$. A system's response to an input is simply the product of the input and the system transfer function. For example, the first term in Eq. 6.3 becomes

$$||D(s)G_{ud}(s)||_2^2 = \mu^2||G_{ud}(s)||_2^2,$$

and the full cost function becomes

$$\begin{aligned} \mathcal{J} = \ &\mu^2||G_{ud}(s)||_2^2 + \mu^2\rho^2||G_{\eta d}(s)||_2^2 \\ &+ ||G_{un}(s)||_2^2 + \rho^2||G_{\eta n}(s)||_2^2. \end{aligned} \tag{6.4}$$

Using the sensitivity expressions in Eq. 6.2, we can write this expression more simply as

$$\mathcal{J} = ||CS||_2^2 + (\mu^2 + \rho^2)||T||_2^2 + \mu^2\rho^2||PS||_2^2. \tag{6.5}$$

6.2 Optimization Method

This section follows Qiu and Zhou's (2013) optimization algorithm. Their cost function in the final equation on page 31 of their book is equivalent to my cost function in Eq. 6.4.

Optimization finds the controller, $C(s)$, that minimizes the cost function. We search for optimal controllers subject to the constraint that all transfer functions in Eq. 6.1 are stable. Stability requires that the real component be negative for all eigenvalues of each transfer function.

A transfer function's eigenvalues are the roots of the denominator's polynomial in s. For each transfer function in Eq. 6.1, the eigenvalues, s, are obtained by solution of $1 + P(s)C(s) = 0$.

We assume a fixed process, P, and weighting coefficients, μ and ρ. To find the optimal controller, we begin with a general form for the controller, such as

$$C(s) = \frac{q_1 s + q_2}{p_0 s^2 + p_1 s + p_2}. \tag{6.6}$$

We seek the coefficients p and q that minimize the cost function.

Qiu and Zhou (2013) solve the example in which $P(s) = 1/s^2$, for arbitrary values of μ and ρ. The accompanying Mathematica code describes the steps in the solution algorithm. Here, I simply state the solution. Check the article by Qiu and Zhou (2013) and my Mathematica code for the details and for a starting point to apply the optimization algorithm to other problems. The following section applies this method to another example and illustrates the optimized system's response to various inputs.

For $P = 1/s^2$, Qiu and Zhou (2013) give the optimal controller

$$C(s) = \frac{\sqrt{2\rho\mu}\left(\sqrt{\rho} + \sqrt{\mu}\right)s + \rho\mu}{s^2 + \sqrt{2}\left(\sqrt{\rho} + \sqrt{\mu}\right)s + \left(\sqrt{\rho} + \sqrt{\mu}\right)^2},$$

with associated minimized cost,

$$\mathcal{J}^* = \sqrt{2}\left[\mu^2\sqrt{\rho} + \rho^2\sqrt{\mu} + 2\rho\mu\left(\sqrt{\mu} + \sqrt{\rho}\right)\right].$$

For $\rho = 1$, the controller becomes

$$C(s) = \frac{\sqrt{2\mu}\left(1 + \sqrt{\mu}\right)s + \mu}{s^2 + \sqrt{2}\left(1 + \sqrt{\mu}\right)s + \left(1 + \sqrt{\mu}\right)^2}, \tag{6.7}$$

with associated minimized cost,

$$\mathcal{J}^* = \sqrt{2}\left[\mu^2 + \sqrt{\mu} + 2\mu(\sqrt{\mu} + 1)\right].$$

We can see the tradeoffs in design most clearly from the controller with $\rho = 1$. When μ is small, load disturbance inputs are smaller than sensor noise inputs. An optimal system should therefore tolerate greater sensitivity to load disturbances in return for reduced sensitivity to sensor noise.

In the optimal controller described by Eq. 6.7, a small value of μ produces low gain, because $C(s)$ becomes smaller as μ declines. We can see from Eq. 6.1 that a small gain for the controller, C, reduces the sensitivity to noise inputs by lowering G_{un} and $G_{\eta n}$. Similarly, a small gain for C raises the sensitivity of the system output, η, to disturbance inputs by raising $G_{\eta d}$.

The optimal system achieves the prescribed rise in sensitivity to disturbance in order to achieve lower sensitivity to noise.

6.3   Resonance Peak Example

This section applies the previous section's $\mathcal{H}_2$ optimization method to the process

$$P(s) = \frac{1}{s^2 + 0.1s + 1}. \tag{6.8}$$

This process has a resonance peak near $\omega = 1$.

My supplemental Mathematica code derives the optimal controller of the form in Eq. 6.6. The optimal controller is expressed in terms of the cost weightings μ and ρ. The solution has many terms, so there is no benefit in showing it here.
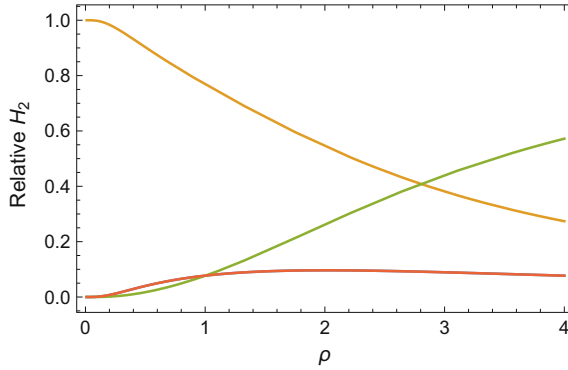
Fig. 6.2 Relative $\mathcal{H}_2$ values for the transfer functions in Eq. 6.1, with $G_{ud} = G_{\eta n}$ in red, $G_{\eta d}$ in gold, and G_{un} in green. The $\mathcal{H}_2$ value for each transfer function is divided by the total $\mathcal{H}_2$ values over all four functions. The transfer functions were derived from the process in Eq. 6.8 and the associated optimal controller. The weighting parameters in the cost function of Eq. 6.4 are $\mu = 1$ and ρ varying along the x-axis of the plot. Swapping values of μ and ρ gives identical results, because of the symmetries in Eqs. 6.1 and 6.4

The general solution in terms of μ and ρ provides a simple way in which to obtain the optimal controller for particular values of μ and ρ. For example, when $\mu = \rho = 1$, the optimal controller is

$$C(s) \approx \frac{0.609(s - 0.81)}{s^2 + 1.73s + 2.49}.$$

Similar controller expressions arise for other values of μ and ρ. Those controllers may be used in the closed loop of Fig. 6.1 to form a complete system.

Figure 6.2 shows the relative $\mathcal{H}_2$ values of the four input–output transfer functions in Eq. 6.1. The $\mathcal{H}_2$ values express sensitivity over all frequencies.

To interpret this figure, look at Eq. 6.4. As the product of the weightings, $\mu\rho$, increases, the output of $G_{\eta d}$ (gold curve) plays an increasingly important role in the total cost relative to the output of G_{un} (green curve).

As the relative cost weighting of $G_{\eta d}$ increases, its $\mathcal{H}_2$ value declines. Similarly, as the relative cost weighting of G_{un} decreases, its $\mathcal{H}_2$ value increases. Once again, we see the sensitivity tradeoffs in response to the relative importance of different perturbations.

The top row of Fig. 6.3 compares the Bode plots for the process, P, and the input–output transfer functions in Eq. 6.1. As ρ increases in the columns from left to right, the rise in the green curve for G_{un} is the strongest change. We can understand that change by examining the cost function in Eq. 6.4. Because $G_{ud} = G_{\eta n}$, a rise in ρ reduces the weighting of G_{un} relative to all other terms.

The strongest increase in relative weighting as ρ rises occurs for $G_{\eta d}$, shown in gold. The mild decline in the gold curve with increasing ρ is consistent with the increased relative cost weighting of that signal.
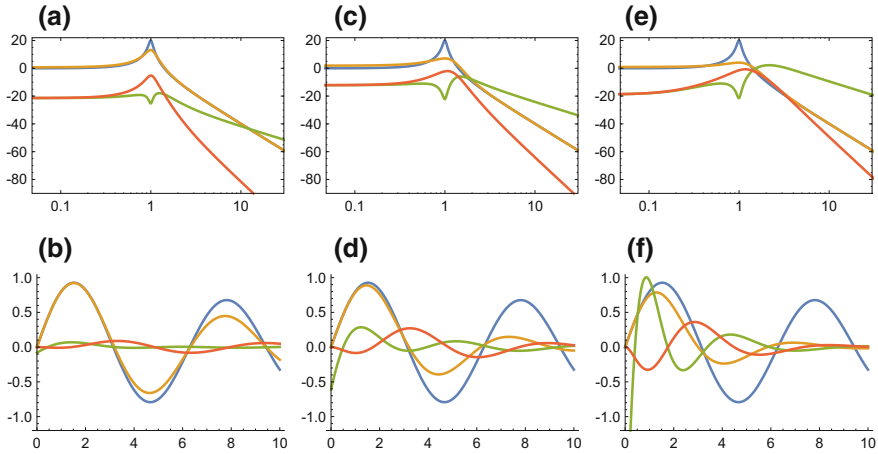
Fig. 6.3 Response of the process in Eq. 6.8 in blue and the transfer functions in Eq. 6.1, with $G_{ud} = G_{\eta n}$ in red, $G_{\eta d}$ in gold, and G_{un} in green. Top row shows Bode magnitude plots. Bottom row shows impulse responses. The input signal weights in Eq. 6.4 are $\mu = 1$ and, for the three columns from left to right, $\rho = 0.25, 1, 4$. Swapping values of μ and ρ gives identical results, because of the symmetries in Eq. 6.1 and 6.4

The bottom row shows the impulse responses. As with the Bode plots, an increase in ρ favors reduced response of $G_{\eta d}$, in gold, causing a smaller impulse response in the right plot with high ρ relative to the left plot with low ρ. Similarly, an increase in ρ weakens the pressure on the G_{un} transfer function in green, causing a larger impulse response with increasing ρ.

6.4 Frequency Weighting

The $\mathcal{H}_2$ norm sums a system's gain over all input frequencies, as in Eq. 5.4. That sum weights all input frequencies equally.

Often, we wish to protect against perturbations that occur primarily in a limited band of frequencies. For example, disturbance loads, d, typically occur at low frequency, reflecting long-term fluctuations or misspecifications in the system's intrinsic processes. In that case, our optimization method should emphasize reducing a system's gain at low frequency with respect to disturbance load inputs and accepting a tradeoff that allows a greater gain at high frequency. By reducing the gain at low frequency, we protect against the common frequencies for load disturbances.

Tradeoffs between low- and high-frequency bands are common. If we start with a process transfer function

$$G(s) = \frac{10(s + 1)}{s + 10},$$

then at zero frequency, $s = j\omega = 0$, the gain is one. As frequency increases, the gain approaches ten.

If we weight this process transfer function by $W(s) = 1/(s + 1)$, then the new system becomes $WG = 10/(s + 10)$. Now, the gain declines with increasing frequency, from a maximum of one at zero frequency to a minimum of zero at infinite frequency.

By weighting the original system, G, by the weighting function, W, we cause the $\mathcal{H}_2$ norm of the combined system, WG, to be relatively more sensitive to low-frequency disturbances. When we design a controller by minimizing the $\mathcal{H}_2$ norm associated with WG, we will typically find a system that is better at rejecting low-frequency load disturbances than a design minimizing the $\mathcal{H}_2$ norm associated with G. For the weighted system, optimization will avoid controllers that reject high-frequency load disturbances, because the weighted system already rejects those high-frequency inputs.

Roughly speaking, a weighting function instructs the optimization method to reduce the gain and sensitivity for certain frequencies and to ignore the gain for other frequencies. The weighting functions do not alter the actual system. The weighting functions are only used to alter the cost function and optimization method that determine the optimal controller.

Figure 6.4 shows the regulation feedback system of Fig. 6.1 with additional weightings for the disturbance and noise inputs. The weightings modify the four system transfer functions and associated sensitivities in Eq. 6.2 to be $W_d G_{ud}$, $W_d G_{\eta d}$, $W_n G_{un}$, and $W_n G_{\eta n}$. The cost function in Eq. 6.5 becomes

$$
\begin{aligned}
\mathcal{J} = \mu^2 ||W_d T||_2^2 + \mu^2 \rho^2 ||W_d P S||_2^2 \\
+ ||W_n C S||_2^2 + \rho^2 ||W_n T||_2^2.
\end{aligned} \tag{6.9}
$$

Consider an example in which we begin with the process, P, in Eq. 6.8. To emphasize low-frequency load disturbances, set $W_d = 1/(s + 0.1)$ to be a low-pass filter. That weighting filters out disturbances that are significantly greater than $\omega = 0.1$. To emphasize high-frequency sensor noise, set $W_n = s/(s + 10)$. That weighting filters out noise that is significantly less than $\omega = 10$. By using these two filters, the optimization method puts very low weight on any disturbances in midrange frequencies of $\omega = (0.1, 10)$.
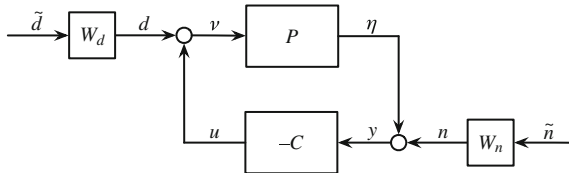
Fig. 6.4 Basic regulation feedback loop in Fig. 6.1 with additional weightings for disturbance and noise inputs. The weightings alter the cost function to emphasize particular frequency bands for disturbance and noise, yielding a modified optimal controller

(a)   **(b)**
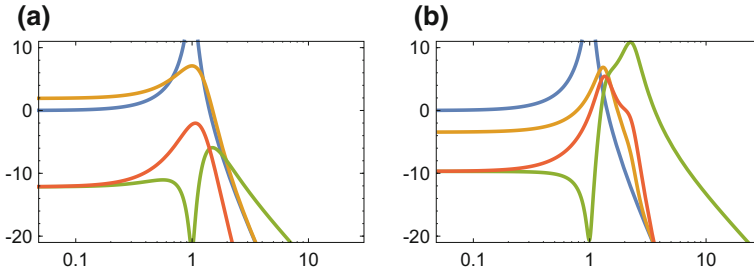
Fig. 6.5 Role of frequency weighted inputs in the design of optimal $\mathcal{H}_2$ controllers for system regulation, illustrated by Bode magnitude plots. **a** Plot for the unweighted case, matching the plot in Fig. 6.3c. **b** Plot for the frequency weighted example in the text, which emphasizes the regulation of low-frequency load disturbances, d, and high-frequency sensor noise, n

By minimizing the weighted $\mathcal{H}_2$ cost in Eq. 6.9, we obtain the optimal controller

$$C(s) = \frac{2.02(s + 1.52)}{s^2 + 1.17s + 6.3}.$$

I calculated the values for this controller by using the numerical minimization function in Mathematica to minimize the $\mathcal{H}_2$ cost, subject to the constraint that all transfer functions in Eq. 6.1 are stable. See the supplemental Mathematica code.

Figure 6.5 compares the optimized system response for the unweighted and weighted cases. Panel **a** shows the Bode magnitude response of the optimized system for the unweighted case, equivalent to the plot in Fig. 6.3c. Panel **b** shows the response of the optimized system for the weighted case in this section.

The weighted case emphasizes low-frequency load disturbances and high-frequency sensor noise, with low weight on midrange frequencies. Comparing the unweighted case in (a) with the weighted case in (b), we see two key differences.

First, the weighted case allows a large rise in magnitudes and associated sensitivity to perturbations for midrange frequencies. That rise occurs because the particular weighting functions in this example discount midrange perturbations.

Second, the gold curve shows that the weighted case significantly reduces the low-frequency sensitivity of system outputs, η, to load disturbances, d. The gold curve describes the response of the transfer function, $G_{\eta d}$. Note that, because of the log scaling for magnitude, almost all of the costs arise in the upper part of the plot. The low relative magnitude for the lower part contributes little to the overall cost.

Chapter 7
Stabilization

The previous chapter assumed that the intrinsic process, P, has a given unvarying form. The actual process may differ from the given form or may fluctuate over time. If a system is designed with respect to a particular form of P, then variation in P away from the assumed form may cause the system to become unstable.

We can take into account the potential variation in P by altering the optimal design problem. The new design problem includes enhanced stability guarantees against certain kinds of variation in P.

Variation in an intrinsic process is an inevitable aspect of design problems. In engineering, the process may differ from the assumed form because of limited information, variability in manufacturing, or fluctuating aspects of the environment.

In biology, a particular set of chemical reactions within an individual may vary stochastically over short time periods. That reaction set may also vary between individuals because of genetic and environmental fluctuations. In all cases, actual processes typically follow nonlinear, time-varying dynamics that often differ from the assumed form.

We may also have variation in the controller or other system processes. In general, how much variability can be tolerated before a stable system becomes unstable? In other words, how robust is a given system's stability to perturbations?

We cannot answer those question for all types of systems and all types of perturbations. However, the $\mathcal{H}_\infty$ norm introduced earlier provides insight for many problems. Recall that the $\mathcal{H}_\infty$ norm is the peak gain in a Bode plot, which is a transfer function's maximum gain over all frequencies of sinusoidal inputs. The small gain theorem provides an example application of the $\mathcal{H}_\infty$ norm.

7.1 Small Gain Theorem

Suppose we have a stable system transfer function, G. That system may represent a process, a controller, or a complex cascade with various feedback loops. To express the mathematical form of G, we must know exactly the dynamical processes of the system.

How much may the system deviate from our assumptions about dynamics and still remain stable? For example, if the uncertainties may be expressed by a positive feedback loop, as in Fig. 7.1, then we can analyze whether a particular system, G, is stably robust against those uncertainties.

In Fig. 7.1, the stable transfer function, Δ, may represent the upper bound on our uncertainty. The feedback loop shows how the nominal unperturbed system, G, responds to an input and becomes a new system, $\tilde{G}$, that accounts for the perturbations. The system, $\tilde{G}$, represents the entire loop shown in Fig. 7.1.

The small gain theorem states that the new system, $\tilde{G}$, is stable if the product of the $\mathcal{H}_\infty$ norms of the original system, G, and the perturbations, Δ, is less than one

$$||G||_\infty ||\Delta||_\infty < 1. \tag{7.1}$$

Here, we interpret G as a given system with a known $\mathcal{H}_\infty$ norm. By contrast, we assume that Δ represents the set of all stable systems that have an $\mathcal{H}_\infty$ norm below some upper bound, $||\Delta||_\infty$. For the perturbed system, $\tilde{G}$, to be stable, the upper bound for the $\mathcal{H}_\infty$ norm of Δ must satisfy

$$||\Delta||_\infty < \frac{1}{||G||_\infty}. \tag{7.2}$$

If G is a system that we can design or control, then the smaller we can make $||G||_\infty$, the greater the upper bound on uncertainty, $||\Delta||_\infty$, that can be tolerated by the perturbed system. Put another way, smaller $||G||_\infty$ corresponds to greater robust stability.
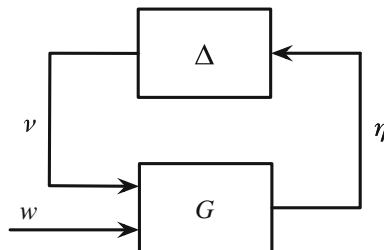
Fig. 7.1 System uncertainty represented by a feedback loop. The transfer function, Δ, describes an upper bound on the extent to which the actual system, $\tilde{G} = G/(1 - G\Delta)$, deviates from the nominal system, G. Here, G may represent a process, a controller, or an entire feedback system

A full discussion of the small gain theorem can be found in textbooks (e.g., Zhou and Doyle 1998; Liu and Yao 2016). I present a brief intuitive summary.

The positive feedback loop in Fig. 7.1 has transfer function

$$\tilde{G} = \frac{G}{1 - G\Delta}. \tag{7.3}$$

We derive that result by the following steps. Assume that the input to G is $w + v$, which is the sum of the external input, w, and the feedback input, v. Thus, the system output is $\eta = G(w + v)$.

We can write the feedback input as the output of the uncertainty process, $v = \Delta\eta$. Substituting into the system output expression, we have

$$\eta = G(w + v) = Gw + G\Delta\eta.$$

The new system transfer function is the ratio of its output to its external input, $\tilde{G} = \eta/w$, which we can solve for to obtain Eq. 7.3.

The new system, $\tilde{G}$, is unstable if any eigenvalue has real part greater than or equal to zero, in which the eigenvalues are the roots of s of the denominator, $1 - G(s)\Delta(s) = 0$.

Intuitively, we can see that $\tilde{G}(s)$ blows up unstably if the denominator becomes zero at some input frequency, ω, for $s = j\omega$. The denominator will be greater than zero as long as the product of the maximum values of $G(j\omega)$ and $\Delta(j\omega)$ are less than one, as in Eq. 7.1. That condition expresses the key idea. The mathematical presentations in the textbooks show that Eq. 7.1 is necessary and sufficient for stability.

Reducing the $\mathcal{H}_\infty$ norm of G increases its robustness with respect to stability. In Eq. 7.2, a smaller $||G||_\infty$ corresponds to a larger upper bound on the perturbations that can be tolerated.

A lower maximum gain also associates with a smaller response to perturbations, improving the robust performance of the system with respect to disturbances and noise. Thus, robust design methods often consider reduction of the $\mathcal{H}_\infty$ norm.

7.2 Uncertainty: Distance Between Systems

Suppose we assume a nominal form for a process, P. We can design a controller, C, in a feedback loop to improve system stability and performance. If we design our controller for the process, P, then how robust is the feedback system to alternative forms of P?

The real process, P', may differ from P because of inherent stochasticity, or because of our simple model for P misspecified the true underlying process.

What is the appropriate set of alternative forms to describe uncertainty with respect to P? Suppose we defined a distance between P and an alternative process, P'. Then

a set of alternatives could be specified as all processes, P', for which the distance from the nominal process, P, is less than some upper bound.

We will write the distance between two processes when measured at input frequency ω as

$$\delta[P(j\omega), P'(j\omega)] = \text{distance at frequency } \omega, \qquad (7.4)$$

for which δ is defined below. The maximum distance between processes over all frequencies is

$$\delta_\nu(P, P') = \max_\omega \; \delta[P(j\omega), P'(j\omega)], \qquad (7.5)$$

subject to conditions that define whether P and P' are comparable (Vinnicombe 2001; Qiu and Zhou 2013). This distance has values $0 \le \delta_\nu \le 1$, providing a standardized measure of separation.

To develop measures of distance, we focus on how perturbations may alter system stability. Suppose we start with a process, P, and controller, C, in a feedback system. How far can an alternative process, P', be from P and still maintain stability in the feedback loop with C? In other words, what is the stability margin of safety for a feedback system with P and C?

Robust control theory provides an extensive analysis of the distances between systems with respect to stability margins (Vinnicombe 2001; Zhou and Doyle 1998; Qiu and Zhou 2010, 2013). Here, I present a rough intuitive description of the key ideas.

For a negative feedback loop with P and C, the various input–output pathways all have transfer functions with denominator $1 + PC$, as in Eq. 6.1. These systems become unstable when the denominator goes to zero, which happens if $P = -1/C$. Thus, the stability margin is the distance between P and $-1/C$.

The values of these transfer functions, $P(j\omega)$ and $C(j\omega)$, vary with frequency, ω. The worst case with regard to stability occurs when P and $-1/C$ are closest; that is, when the distance between these functions is a minimum with respect to varying frequency. Thus, we may define the stability margin as the minimum distance over frequency

$$b_{P,C} = \min_\omega \; \delta[P(j\omega), -1/C(j\omega)]. \qquad (7.6)$$

Here is the key idea. Start with a nominal process, P_1, and a controller, C. If an alternative or perturbed process, P_2, is close to P_1, then the stability margin for P_2 should not be much worse than for P_1.

In other words, a controller that stabilizes P_1 should also stabilize all processes that are reasonably close to P_1. Thus, by designing a good stability margin for P_1, we guarantee robust stabilization for all processes sufficiently near P_1.

We can express these ideas quantitatively, allowing the potential to design for a targeted level of robustness. For example,

$$b_{P_2,C} \ge b_{P_1,C} - \delta_\nu(P_1, P_2).$$

Read this as the guaranteed stability margin for the alternative process is at least as good as the stability margin for nominal process minus the distance between the nominal and alternative processes. A small distance between processes, δ_ν, guarantees that the alternative process is nearly as robustly stable as the original process.

The definitions in this section depend on the distance measure, expressed as

$$\delta(c_1, c_2) = \frac{|c_1 - c_2|}{\sqrt{1 + |c_1|^2}\sqrt{1 + |c_2|^2}}.$$

Here, c_1 and c_2 are complex numbers. Transfer functions return complex numbers. Thus, we can use this function to evaluate $\delta[P_1(j\omega), P_2(j\omega)]$.

7.3   Robust Stability and Robust Performance

The stability margin $b_{P,C}$ measures the amount by which P may be altered and still allow the system to remain stable. Note that $b_{P,C}$ in Eq. 7.6 expresses a minimum value of δ over all frequencies. Thus, we may also think of $b_{P,C}$ as the maximum value of $1/\delta$ over all frequencies.

The maximum value of magnitude over all frequencies matches the definition of the $\mathcal{H}_\infty$ norm, suggesting that maximizing the stability margin corresponds to minimizing some expression for an $\mathcal{H}_\infty$ norm. Indeed, there is such an $\mathcal{H}_\infty$ norm expression for $b_{P,C}$. However, the particular form is beyond our scope. The point here is that robust stability via maximization of $b_{P,C}$ falls within the $\mathcal{H}_\infty$ norm theory, as in the small gain theorem.

Stability is just one aspect of design. Typically, a stable system must also meet other objectives, such as rejection of disturbance and noise perturbations. This section shows that increasing the stability margin has the associated benefit of improving a system's rejection of disturbance and noise. Often, a design that targets reduction of the $\mathcal{H}_\infty$ norm gains the benefits of an increased stability margin and better regulation through rejection of disturbance and noise.

The previous section on regulation showed that a feedback loop reduces its response to perturbations by lowering its various sensitivities, as in Eqs. 6.2 and 6.5. A feedback loop's sensitivity is $S = 1/(1 + PC)$ and its complementary sensitivity is $T = PC/(1 + PC)$.

Increasing the stability margin, $b_{P,C}$, reduces a system's overall sensitivity. We can see the relation between stability and sensitivity by rewriting the expression for $b_{P,C}$ as

$$b_{P,C} = \left[\max_\omega \sqrt{|S|^2 + |CS|^2 + |PS|^2 + |T|^2}\right]^{-1}$$

This expression shows that increasing $b_{P,C}$ reduces the total magnitude of the four key sensitivity measures for negative feedback loops.
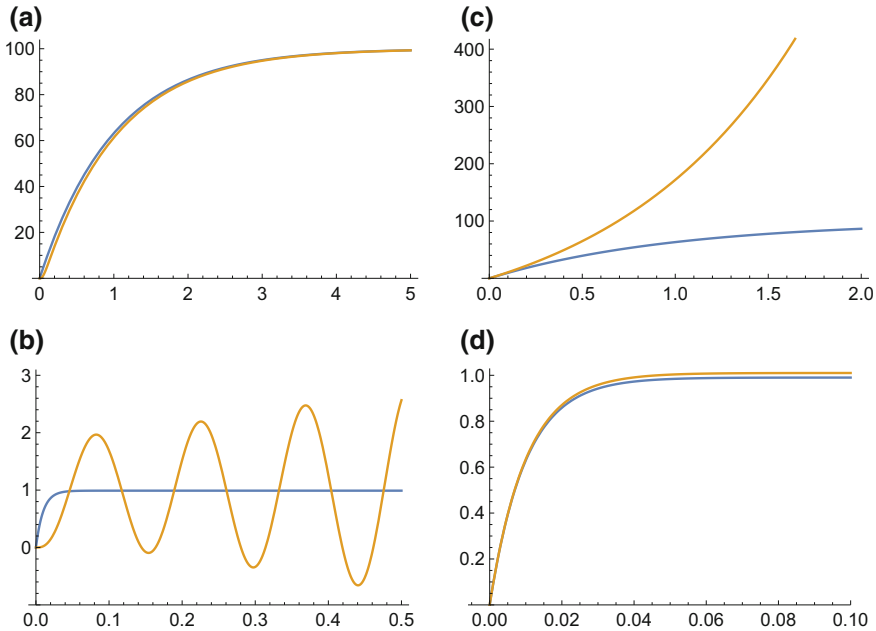
Fig. 7.2 Comparison between the responses of two systems to a unit step input, $r = 1$. The blue curves show P_1 and the gold curves show P_2. **a, b** Systems in Eq. 7.7, with $k = 100$ and $T = 0.025$. The top plot shows the open-loop response for each system. The bottom plot shows the closed-loop feedback response with unit feedback, $P/(1 + P)$, in which the error signal into the system, P, is $1 - y$ for system output, y. **c, d** Open (top) and closed (bottom) loop responses for the systems in Eq. 7.8, with $k = 100$. Redrawn from Fig. 12.3 of Åström and Murray (2008), ©Princeton University Press

7.4 Examples of Distance and Stability

The measure, $\delta_\nu(P_1, P_2)$, describes the distance between processes with respect to their response characteristics in a negative feedback loop. The idea is that P_1 and P_2 may have different response characteristics when by themselves in an open loop, yet have very similar responses in a feedback loop. Or P_1 and P_2 may have similar response characteristics when by themselves, yet have very different responses in a feedback loop.

Thus, we cannot simply use the response characteristics among a set of alternative systems to understand how variations in a process influence stability or performance. Instead, we must use a measure, such as δ_ν, that expresses how variations in a process affect feedback loop characteristics.

This section presents two examples from Sect. 12.1 of Åström and Murray (2008). In the first case, the following two systems have very similar response characteristics by themselves in an open loop, yet have very different responses in a closed feedback loop

$$P_1 = \frac{k}{s+1} \qquad P_2 = \frac{k}{(s+1)(Ts+1)^2}, \qquad (7.7)$$

when evaluated at $k = 100$ and $T = 0.025$, as shown in Fig. 7.2a, b. The distance between these systems is $\delta_\nu(P_1, P_2) = 0.89$. That large distance corresponds to the very different response characteristics of the two systems when in a closed feedback loop. (Åström and Murray (2008) report a value of $\delta_\nu = 0.98$. The reason for the discrepancy is not clear. See the supplemental Mathematica code for my calculations, derivations, and graphics here and throughout the book.)

In the second case, the following two systems have very different response characteristics by themselves in an open loop, yet have very similar responses in a closed feedback loop

$$P_1 = \frac{k}{s+1} \qquad P_2 = \frac{k}{s-1}, \qquad (7.8)$$

when evaluated at $k = 100$, as shown in Fig. 7.2c, d. The distance between these systems is $\delta_\nu(P_1, P_2) = 0.02$. That small distance corresponds to the very similar response characteristics of the two systems when in a closed feedback loop.

7.5 Controller Design for Robust Stabilization

The measure $b_{P,C}$ describes the stability margin for a feedback loop with process P and controller C. A larger margin means that the system remains robustly stable to variant processes, P', with greater distance from the nominal process, P. In other words, a larger margin corresponds to robust stability against a broader range of uncertainty.

For a given process, we can often calculate the controller that provides the greatest stability margin. That optimal controller minimizes an $\mathcal{H}_\infty$ norm, so in this case we may consider controller design to be an $\mathcal{H}_\infty$ optimization method.

Often, we also wish to keep the $\mathcal{H}_2$ norm small. Minimizing that norm improves a system's regulation by reducing response to perturbations. Jointly optimizing the stability margin and rejection of disturbances leads to mixed $\mathcal{H}_\infty$ and $\mathcal{H}_2$ design.

Mixed $\mathcal{H}_\infty$ and $\mathcal{H}_2$ optimization is an active area of research (Chen and Zhou 2001; Chang 2017). Here, I briefly summarize an example presented in Qiu and Zhou (2013). That article provides an algorithm for mixed optimization that can be applied to other systems.

Qiu and Zhou (2013) start with the process, $P = 1/s^2$. They consider three cases. First, what controller provides the minimum $\mathcal{H}_\infty$ norm and associated maximum stability margin, b, while ignoring the $\mathcal{H}_2$ norm? Second, what controller provides the minimum $\mathcal{H}_2$ norm, while ignoring the stability margin and $\mathcal{H}_\infty$ norm? Third, what controller optimizes a combination of the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms?

For the first case, the controller

$$C(s) = \frac{\left(1 + \sqrt{2}\right)s + 1}{s + 1 + \sqrt{2}}$$

has the maximum stability margin

$$b_{P,C}^* = \left(4 + 2\sqrt{2}\right)^{-1/2} = 0.38.$$

The cost associated with the $\mathcal{H}_2$ norm from Eq. 6.5 is $\mathcal{I} = \infty$, because the sensitivity function CS has nonzero gain at infinite frequency.

For the second case, the controller

$$C(s) = \frac{2\sqrt{2}s + 1}{s^2 + 2\sqrt{2}s + 4}$$

has the minimum $\mathcal{H}_2$ cost, $\mathcal{I}^* = 6\sqrt{2} = 8.49$, with associated stability margin $b_{P,C} = 0.24$. This controller and associated cost match the earlier example of $\mathcal{H}_2$ norm minimization in Eq. 6.7 with $\mu = 1$.

For the third case, we constrain the minimum stability margin to be at least $b_{P,C} > 1/\sqrt{10} = 0.316$, and then find the controller that minimizes the $\mathcal{H}_2$ norm cost subject to the minimum stability margin constraint, yielding the controller

$$C(s) = \frac{2.5456s + 1}{0.28s^2 + 1.5274s + 2.88},$$

which has the cost $\mathcal{I} = 13.9$ and stability margin $b_{P,C} = 0.327$.

In these examples, a larger stability margin corresponds to a greater $\mathcal{H}_2$ cost. That relation illustrates the tradeoff between robust stability and performance measured by the rejection of disturbance and noise perturbations.

Chapter 8
Tracking

The previous chapters on regulation and stabilization ignored the reference input, r. In those cases, we focused on a system's ability to reject perturbations and to remain stable with respect to uncertainties. However, a system's performance often depends strongly on its ability to track external environmental or reference signals.

To study tracking of a reference input, let us return to the basic feedback loop structure in Fig. 2.1c, shown again in Fig. 8.1. Good tracking performance means minimizing the error, $e = r - y$, the difference between the reference input and the system output.

Typically, we can reduce tracking error by increasing the control signal, u, which increases the speed at which the system changes its output to be closer to the input. However, in a real system, a larger control signal requires more energy. Thus, we must consider the tradeoff between minimizing the error and reducing the cost of control.

I previously introduced a cost function that combines the control and error signals in Eq. 5.1 as

$$\mathcal{J} = \int_0^T (u^2 + \rho^2 e^2)\mathrm{d}t, \tag{8.1}$$

in which $u(t)$ and $e(t)$ are functions of time, and ρ is a weighting for the relative importance of the error signal relative to the control signal.

I noted in Eq. 5.2 that the square of the $\mathcal{H}_2$ norm is equal to the energy of a signal, for example,

$$\|e(t)\|_2^2 = \int_0^\infty |e(t)|^2 \mathrm{d}t.$$

In this chapter, we will consider reference signals that change over time. A system will typically not track a changing reference perfectly. Thus, the error will not go to zero over time, and the energy will be infinite. For infinite energy, we typically cannot use the $\mathcal{H}_2$ norm. Instead, we may consider the average of the squared signal per unit time, which is the power. Or we may analyze the error over a finite time period, as in Eq. 8.1.

Fig. 8.1 System with a basic
feedback loop in response to
a reference input, r
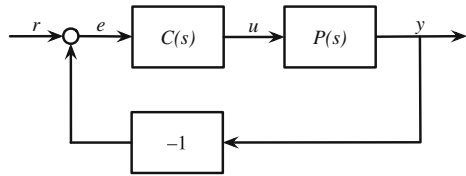
To analyze particular problems, we begin by expressing the transfer function for
the error from Eq. 3.5 as

$$E(s) = R(s) - Y(s) = \frac{1}{1 + C(s)P(s)} R(s).$$

We may write the transfer function for the control signal as

$$U(s) = C(s)E(s) = \frac{C(s)}{1 + C(s)P(s)} R(s).$$

These equations express the key tradeoff between the error signal and the control
signal. A controller, C, that outputs a large control signal reduces the error, E, and
increases the control signal, U. The following example illustrates this tradeoff and
the potential consequences for instability.

8.1 Varying Input Frequencies

To analyze the cost over a particular time period, as in Eq. 8.1, we must express the
transfer functions as differential equations that describe change over time. We can
use the basic relation between transfer functions in Eq. 2.5 and differential equations
in Eq. 2.6.

In this example, I use the process in Eq. 4.1 that I analyzed in earlier chapters

$$P(s) = \frac{1}{(s + 0.1)(s + 10)}.$$

I use the controller

$$C(s) = \frac{q_0 s^2 + q_1 s + q_2}{p_0 s^2 + p_1 s + p_2}. \tag{8.2}$$

Our goal is to find a controller of this form that minimizes the cost function in Eq. 8.1.

I use a reference signal that is the sum of three sine waves with frequencies
$\omega_i = (\psi^{-1}, 1, \psi)$. I weight each frequency by $\kappa_i = (1, 1, 0.2)$, such that the high
frequency may be considered a rapid, relatively low-amplitude disturbance. Thus,
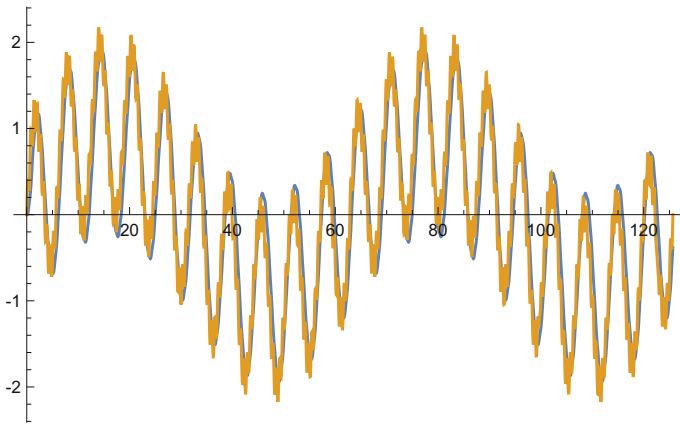
Fig. 8.2 Reference signal, r, in gold, from Eq. 8.3, and the filtered signal, r_w, in blue, from the filter in Eq. 8.4 applied to the reference signal. The blue curves in Fig. 8.3 show the filtered signal more clearly

$$R(s) = \sum_i \frac{\kappa_i \omega_i}{s^2 + \omega_i^2}, \tag{8.3}$$

in which each of the three terms in the sum expresses a sine wave with frequency ω_i. Here, I use $\psi = 10$.

Often, low-frequency signals represent true changes in the external environment. By contrast, high-frequency inputs represent noise or signals that change too rapidly to track effectively. Thus, we may wish to optimize the system with respect to low-frequency inputs and to ignore high-frequency inputs.

We can accomplish frequency weighting by using a filtered error signal in the cost function, $E_W(s) = R(s)W(s) - Y(s)$, for a weighting function W that passes low frequencies and reduces the gain of high frequencies. The weighted error signal as a function of time is $e_w(t)$.

In our example, the function

$$W(s) = \left(\frac{\sqrt{\psi}}{s + \sqrt{\psi}} \right)^3 \tag{8.4}$$

will reduce the relative weighting of the high-frequency input at frequency ψ. I use the filtered error signal, e_w, for the cost function in Eq. 8.1, yielding

$$\mathcal{J} = \int_0^T (u^2 + \rho^2 e_w^2) \mathrm{d}t. \tag{8.5}$$

The gold curve in Fig. 8.2 shows the environmental reference signal, r, for the associated transfer function, $R(s)$. The blue curve shows the filtered reference signal, r_w, for the filtered system, $R(s)W(s)$. The filtered curve removes the high-frequency
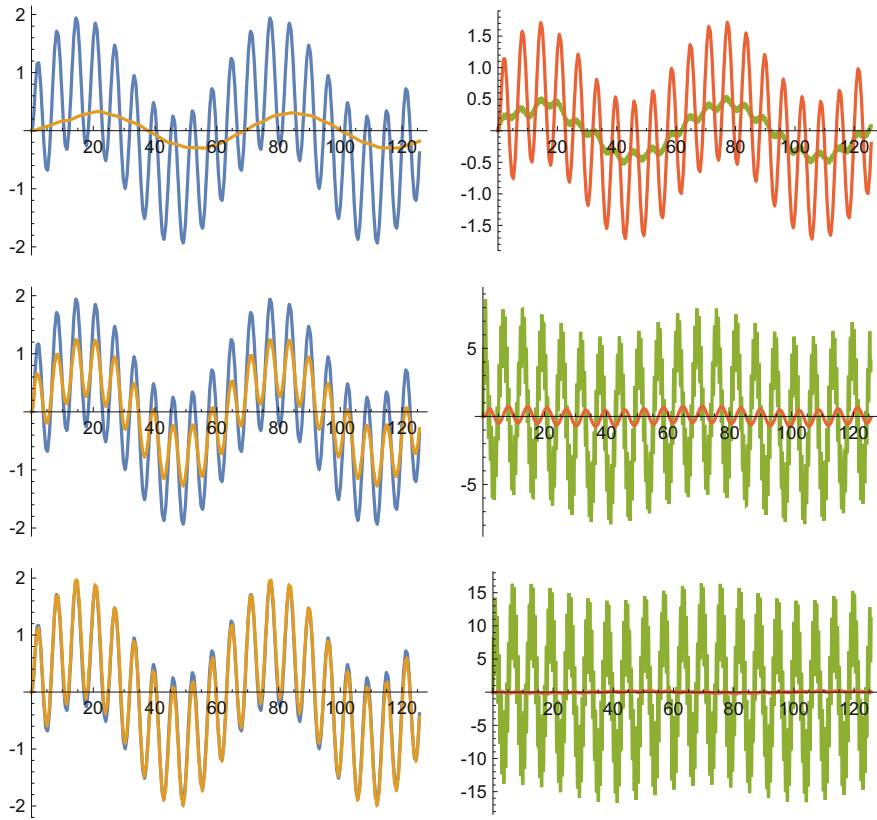
Fig. 8.3 Optimization of the cost function in Eq. 8.5 for the controller in Eq. 8.2. The left column shows the tracking performance. The blue curve traces the filtered reference signal, r_w, associated with $R(s)W(s)$. The gold curve traces the system output, y, associated with $Y(s)$. The difference between the curves is the error, $e_w = r_w - y$. The right column shows the error, e_w, in red, and the control signal, u for $U(s)$, in green. The rows show, from top to bottom, an increased weighting of the error versus the control signal in the cost, $\mathcal{J}$, in Eq. 8.5, with $\rho = (1, 10, 100)$. The optimized controllers may represent local rather than global optima. See the supplemental Mathematica code

noise of the reference signal and closely matches the fluctuations from the two lower frequency sine wave inputs.

Figure 8.3 illustrates the tradeoff between the tracking performance and the cost of the control signal energy to drive the system. The cost function in Eq. 8.5 describes the tradeoff between tracking, measured by the squared error between the filtered reference signal and the system output, e_w^2, and the control signal energy, u^2.

The parameter ρ sets the relative balance between these opposing costs. A higher ρ value favors closer tracking and smaller error because a high value of ρ puts less weight on the cost of the control signal. With a lower cost for control, the controller can output a stronger signal to drive the system toward a closer match with the target reference signal.

8.2 Stability Margins

Minimizing a quadratic cost function or an $\mathcal{H}_2$ norm may lead to a poor stability margin. For example, close tracking of a reference signal may require a large control signal from the controller. Such high gain feedback creates rapidly responding system dynamics, which can be sensitive to uncertainties.

In Fig. 8.3, the stability margins for the three rows associated with $\rho = (1, 10, 100)$ are $b_{P,C} = (0.285, 0.023, 0.038)$. A robust stability margin typically requires a value greater than approximately 1/3 or perhaps 1/4.

In this case, the system associated with $\rho = 1$ has a reasonable stability margin, whereas the systems associated with higher ρ have very poor stability margins. The poor stability margins suggest that those systems could easily be destabilized by perturbations of the underlying process or controller dynamics.

We could minimize the cost function subject to a constraint on the lower bound of the stability margin. However, numerical minimization for that problem can be challenging. See the supplemental Mathematica code for an example.

Chapter 9
State Feedback

A transfer function corresponds to a time-invariant, linear system of ordinary differential equations. In an earlier chapter, I showed the general form of a transfer function in Eq. 2.5 and the underlying differential equations in Eq. 2.6.

For example, the transfer function $P(s) = 1/(s + a)$ with input u and output y corresponds to the differential equation $\dot{x} = -ax + u$, with output $y = x$. Here, x is the internal state of the process. Models that work directly with internal states are called state-space models.

Transfer functions provide significant conceptual and analytical advantages. For example, the multiplication of transfer functions and the simple rules for creating feedback loops allow easy creation of complex process cascades. With regard to system response, a Bode plot summarizes many aspects in a simple, visual way.

However, it often makes sense to analyze the underlying states directly. Consider, for example, the regulation of an organism's body temperature. We could model performance and cost in terms of body temperature. Alternatively, the underlying states may include the burning of stored energy, the rise and fall of various signaling molecules, the dilation of blood vessels, and so on.

Direct analysis of those internal states provides advantages. The individual states may have associated costs, which we could study directly in our cost function. We could consider the regulatory control of the individual states rather than temperature because temperature is an aggregate outcome of the underlying states. For example, each state could be regulated through feedback, in which the feedback into one state may depend on the values of all of the states, allowing more refined control of costs and performance.

When we use a state-space analysis, we do not have to give up all of the tools of frequency analysis that we developed for transfer functions. For example, we can consider the response of a system to different input frequencies.

State-space models can also describe time-varying, nonlinear dynamics. The response of a nonlinear system will change with its underlying state, whereas transfer function systems have a constant frequency response.

9.1   Regulation Example

In the prior chapter on regulation, I analyzed the process in Eq. 6.8 as

$$P(s) = \frac{1}{s^2 + \alpha s + \beta},\tag{9.1}$$

with $\alpha = 0.1$ and $\beta = 1$. This process has a resonance peak near $\omega = 1$. The state-space model for this process is

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\beta x_1 - \alpha x_2 + u \\ y &= x_1,\end{aligned}\tag{9.2}$$

in which the dynamics are equivalent to a second-order differential equation, $\ddot{x} + \alpha\dot{x} + \beta x = u$, with $y = x$.

For a state-space regulation problem, the design seeks to keep the states close to their equilibrium values. We can use equilibrium values of zero without loss of generality. When the states are perturbed away from their equilibrium, we adjust the input control signal, u, to drive the states back to their equilibrium.

The cost function combines the distance from equilibrium with regard to the state vector, $\mathbf{x}$, and the energy required for the control signal, $\mathbf{u}$. Distances and energies are squared deviations from zero, which we can write in a general way in vector notation as

$$\mathcal{J} = \int_0^T \left(\mathbf{u}'\mathbf{R}\mathbf{u} + \mathbf{x}'\mathbf{Q}\mathbf{x}\right)\mathrm{d}t,\tag{9.3}$$

in which $\mathbf{R}$ and $\mathbf{Q}$ are matrices that give the cost weightings for components of the state vector, $\mathbf{x} = x_1, x_2, \ldots$, and components of the input vector, $\mathbf{u} = u_1, u_2, \ldots$. In the example here, there is only one input. However, state-space models easily extend to handle multiple inputs.

For the regulation problem in Fig. 9.1, the goal is to find the feedback gains for the states given in the matrix $\mathbf{K}$ that minimize the cost function. The full specification of the problem requires the state equation matrices for use in Eq. 2.6, which we have from Eq. 9.2 as

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\beta & -\alpha \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} 1 & 0 \end{pmatrix},\tag{9.4}$$

and the cost matrices, $\mathbf{R}$ and $\mathbf{Q}$. In this case, we have a single input, so the cost matrix for inputs, $\mathbf{R}$, can be set to one, yielding an input cost term, u^2.

For the state costs, we could ignore the second state, x_2, leaving only $x_1 = y$, so that the state cost would be proportional to the squared output, $y^2 = e^2$. Here, y is
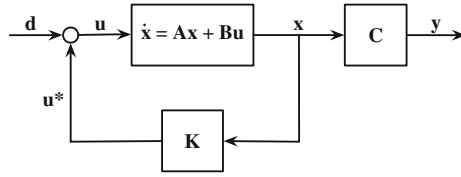
Fig. 9.1 State feedback model of regulation. The process and output describe the state equations in Eq. 2.6. The control input signal, $\mathbf{u}^* = \mathbf{Kx}$, is obtained by minimizing the cost function in Eq. 9.3 to derive the optimal state gains. A disturbance, $\mathbf{d}$, is added to the input signal

equivalent to the error, $e = y - r$, because the reference input is $r = 0$. A cost based on u^2 and e^2 matches the earlier cost function in Eq. 8.1.

In this case, I weight the costs for each state equally by letting $\mathbf{Q} = \rho^2 \mathbf{I}_2$, in which $\mathbf{I}_n$ is the identity matrix of dimension n, and ρ is the cost weighting for states relative to inputs. With those definitions, the cost becomes

$$\mathcal{I} = \int_0^T \left[u^2 + \rho^2 (x_1^2 + x_2^2) \right] dt,$$

in which $x_1^2 + x_2^2$ measures the distance of the state vector from the target equilibrium of zero.

We obtain the gain matrix for state feedback models, $\mathbf{K}$, by solving a matrix Riccati equation. Introductory texts on control theory derive the Riccati equation. For our purposes, we can simply use a software package, such as Mathematica, to obtain the solution for particular problems. See the supplemental software code for an example.

Figure 9.2 shows the response of the state feedback system in Fig. 9.1 with the Riccati solution for the feedback gain values, $\mathbf{K}$. Within each panel, the different curves show different values of ρ, the ratio of the state costs for $\mathbf{x}$ relative to the input costs for $\mathbf{u}$. In the figure, the blue curves show $\rho = 1/4$, which penalizes the input costs four times more than the state costs. In that case, the control inputs tend to be costly and weaker, allowing the state values to be larger.

At the other extreme, the green curves show $\rho = 4$. That value penalizes states more heavily and allows greater control input values. The larger input controls drive the states back toward zero much more quickly. The figure caption provides details about each panel.

In this example, the underlying equations for the dynamics do not vary with time. Time-invariant dynamics correspond to constant values in the state matrices, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. A time-invariant system typically leads to constant values in the optimal gain matrix, $\mathbf{K}$, obtained by solving the Riccati equation.

The Riccati solution also works when those coefficient matrices have time-varying values, leading to time-varying control inputs in the optimal gain matrix, $\mathbf{K}$. The general approach can also be extended to nonlinear systems. However, the Riccati equation is not sufficient to solve nonlinear problems.
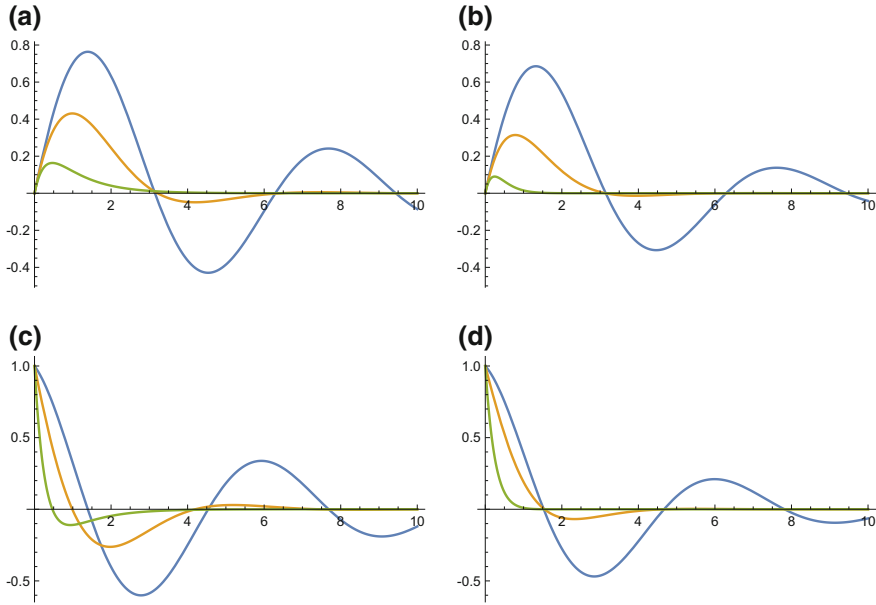
Fig. 9.2 Response to impulse perturbations of systems with state feedback, as in Fig. 9.1. **a** Response of the state-space system in Eq. 9.2. Curves show $x_1 = y$ for cost ratio $\rho = (0.25, 1, 4)$ in blue, gold, and green, respectively. In this case, the impulse perturbation enters the system through u in Eq. 9.2, affecting $\dot{x}_2$. **b** Modified state-space model that has two inputs, one each into $\dot{x}_1$ and $\dot{x}_2$, associated with the state matrix $\mathbf{B} = \mathbf{I}_2$. Impulse perturbation comes into $\dot{x}_2$ as in the original model. In this case, there are two control inputs for feedback via the gain matrix, $\mathbf{K}$. The optimization uses both inputs, allowing the feedback to control each state separately. That extension of control directly to all states allows the feedback system to bring the state responses back to zero more quickly than in the original system with only one state feedback. **c** and **d** Response of the second state, x_2. Systems for each panel match to the corresponding panels above. Note in **d** that the second input for feedback drives the state to zero more quickly than in **c**, which has only one input

Methods that minimize quadratic costs or $\mathcal{H}_2$ norms can produce systems with poor stability margins. To obtain guaranteed stability margins, one can minimize costs subject to a constraint on the minimum stability margin.

9.2   Tracking Example

Consider the tracking example from the previous chapter. That example began with the process in Eq. 4.1 as

$$P(s) = \frac{1}{(s+a)(s+b)} = \frac{1}{s^2 + \alpha s + \beta},$$

with $\alpha = a + b = 10.1$ and $\beta = ab = 1$. The state-space model is given in Eq. 9.2, expressed in matrix form in Eq. 9.4. The state-space model describes the process output over time, $y(t)$, which we abbreviate as y.

Here, I describe a state-space design of tracking control for this process. For this example, I use the tracking reference signal in Eq. 8.3, ignoring high-frequency noise ($\kappa_2 = 0$). The reference signal is the sum of low-frequency ($\omega_0 = 0.1$) and mid-frequency ($\omega_1 = 1$) sine waves. The transfer function for the reference signal is

$$R(s) = \frac{\omega_0}{s^2 + \omega_0^2} + \frac{\omega_1}{s^2 + \omega_1^2}.$$

In state-space form, the reference signal, $r(t)$, is

$$\mathbf{A_r} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\omega_0^2\omega_1^2 & 0 & -\omega_0^2 - \omega_1^2 & 0 \end{pmatrix}$$

$$\mathbf{B_r} = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^\mathsf{T}$$

$$\mathbf{C_r} = \begin{pmatrix} \omega_0^2\omega_1 + \omega_0\omega_1^2 & 0 & \omega_0 + \omega_1 & 0 \end{pmatrix}.$$

We can transform a tracking problem into a regulator problem and then use the methods from the previous chapter (Anderson and Moore 1989). In the regulator problem, we minimized a combination of the squared inputs and states. For a tracking problem, we use the error, $e = y - r$, instead of the state values, and express the cost as

$$\mathcal{J} = \int_0^T \left(\mathbf{u'Ru} + e^2 \right) \mathrm{d}t. \tag{9.5}$$

We can combine the state-space expressions for y and r into a single state-space model with output e. That combined model allows us to apply the regulator theory to solve the tracking problem with state feedback.

The combined model for the tracking problem is

$$\mathbf{A_t} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_r} \end{pmatrix}$$

$$\mathbf{B_t} = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{B_r} \end{pmatrix}$$

$$\mathbf{C_t} = \begin{pmatrix} \mathbf{C} & -\mathbf{C_r} \end{pmatrix},$$

which has output determined by $\mathbf{C_t}$ as $e = y - r$ (Anderson and Moore 1989). In this form, we can apply the regulator theory to find the optimal state feedback matrix, $\mathbf{K}$,

(a)

(b)

(c)

(d)
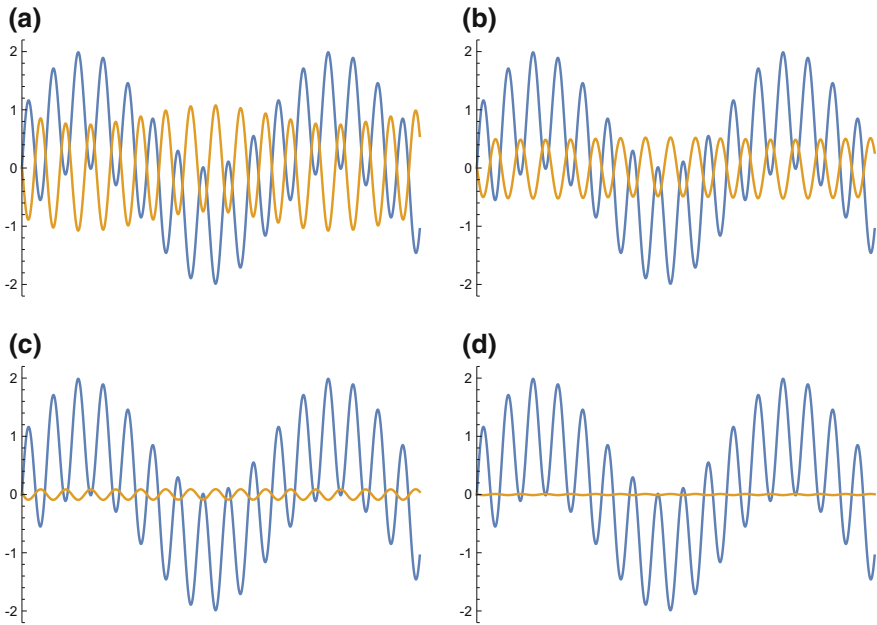
Fig. 9.3 Tracking a reference input signal with state feedback. The blue curve shows the input signal, $r(t)$, which is the sum of two sine waves with frequencies $\omega_0 = 0.1$ and $\omega_1 = 1$. The system responds to the input by producing an output, $y(t)$. The output is determined by the process, $P(s)$, and the optimal state feedback, **K**, as presented in the text. The gold curves show the system error, which is $e = y - r$, the difference between the output and the reference signal. **a** Squared input values are weighted by $\mathbf{R} = w\mathbf{I}_n$, with $w = 0.1$ and n as the number of inputs to the process. In this case, we fix the input to the embedded reference signal in the state-space model to zero and have one input into the process given by **B** in Eq. 9.4. The error curve shows that this system closely tracks the low-frequency reference sine wave but does not track the high-frequency reference component. **b** This case allows feedback inputs into both states of the process, augmenting $\dot{x}_1$ in Eq. 9.2 with a separate input and letting $\mathbf{B} = \mathbf{I}_2$. Other aspects as in the prior panel. **c** As in panel **a**, with $w = 0.01$. The weaker cost for inputs allows stronger feedback inputs and closer tracking of the high-frequency component of the reference signal, thus shrinking the tracking error in the gold curve. **d** Nearly perfect tracking with $w = 0.01$ and inputs directly into both process states. See supplemental Mathematica code for details about assumptions and calculations

that minimizes the costs, $\jmath$, in Eq. 9.5. Figure 9.3 presents an example and mentions some technical issues in the caption.

The example illustrates two key points. First, as the relative cost weighting of the inputs declines, the system applies stronger feedback inputs and improves tracking performance.

Second, the state equations for the intrinsic process, $P(s)$, in Eq. 9.4 provide input only into the second state of the process, as can be seen in the equation for $\dot{x}_2$ in Eq. 9.2. When we allow a second input into the intrinsic process, $P(s)$, by allowing feedback directly into both $\dot{x}_1$ and $\dot{x}_2$, we obtain much better tracking performance, as shown in Fig. 9.3.

Part III
Common Challenges

Parts I and II presented the fundamental principles of linear control theory. Part III extends the core theory. The extensions introduce common problems in application and potential solutions. These problems are active topics in current research.

Chapter 10
Nonlinearity

Real systems are nonlinear. Before discussing nonlinear theory, I review three reasons why the core theory of control focuses on linear analysis.

First, feedback compensates for model uncertainty. Suppose we analyze a feedback system based on a linear model of dynamics, and the true model is nonlinear. If the linear model captures essential aspects of the dynamics sufficiently, the true nonlinear feedback system will often have the same qualitative response characteristics as the modeled linear system. As Vinnicombe (2001, p. xvii) emphasized: "One of the key aims of using feedback is to minimize the effects of lack of knowledge about a system which is to be controlled."

Second, the fundamental principles of control systems apply to both linear and nonlinear dynamics. The comprehensive theory for linear systems provides insight into nonlinear systems. For example, strong feedback signals often help to minimize error but can create instabilities. Controllers can be added at different points in a system to filter signals or modulate inputs that drive processes. Primary goals of analysis and design often emphasize stability, disturbance rejection, regulation or tracking. Certain tradeoffs inevitably arise. Integral control smooths response toward a long-term target. Derivative control improves the speed of response by using a simple prediction.

Third, the main tools for the analysis and design of nonlinear systems typically extend the tools developed for linear systems. For example, nonlinear systems are approximately linear near a particular operating point. One can study the linear approximation around that point, and then switch to an alternative linear approximation as the system moves to another operating domain. By piecing together the different linear approximations in neighboring domains, one develops a sequence of linear systems that together capture much of the nonlinear characteristics. Other tools of nonlinear analysis typically leverage the deep insights and methods of linear analysis (Slotine and Li 1991; Isidori 1995; Khalil 2002).

This chapter presents a few brief illustrations of nonlinear control systems.

10.1   Linear Approximation

In cellular biology, the concentration of a molecule, m, often responds an input signal, ϕ, by the Hill equation

$$m = k\Big(\frac{\phi^n}{1 + \phi^n}\Big),$$

with parameters k and n. In this example, I let $k = 1$.

The nonlinear reaction system

$$\dot{x}_1 = 1 - x_1 + u$$
$$\dot{x}_2 = \frac{x_1^n}{1 + x_1^n} - \gamma x_2 \tag{10.1}$$

describes the change in the output, $y = x_2$, as driven by a Hill equation response to x_1 and an intrinsic decay of x_2 at a rate γ. In the absence of external input, u, the internal dynamics hold the concentration of x_1 at the equilibrium value of $x_1^* = 1$, which in turn sets the equilibrium of the output at $x_2^* = 1/2\gamma$. The system responds to external control signals or perturbations through u, which drives the concentration of x_1, which in turn drives the concentration of x_2.

We can study the dynamics and control of this system by linearizing the dynamics near a particular operating point, $(\hat{x}_1, \hat{x}_2)$. In this example, we use as the operating point the equilibrium, (x_1^*, x_2^*), in the absence of external input, $u = 0$. The linearized system expresses the deviations from the equilibrium point as

$$\dot{x}_1 = -x_1 + u$$
$$\dot{x}_2 = nx_1/4 - \gamma x_2. \tag{10.2}$$

Figure 10.1 shows the response to an impulse perturbation by the nonlinear system and the linear approximation. In the left panel, with a weak impulse and small deviation from the equilibrium, the nonlinear and linear dynamics are nearly identical. As the impulse becomes stronger in the right panels, the deviation from equilibrium increases and the dynamics of the linear approximation diverge from the original linear system.

10.2   Regulation

We can analyze the benefits of feedback for regulating nonlinear dynamics. One approach analyzes feedback for the linear approximation near the target equilibrium. The feedback for the linear approximation should provide good regulation when applied to the nonlinear system near the equilibrium.
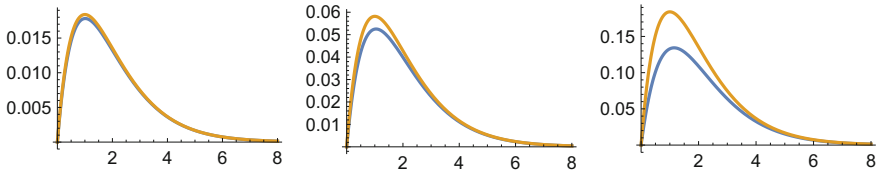
Fig. 10.1 Response to an impulse perturbation by a nonlinear system and a linearized approximation, shown as the deviation from equilibrium. The nonlinear response in blue arises from the system in Eq. 10.1. The linearized response in gold arises from the system in Eq. 10.2. The panels from left to right show increasing magnitudes of the Dirac delta impulse perturbation at time zero, with the impulse weighted by $0.1, 0.1\sqrt{10}, 1$, respectively. Larger impulses cause greater deviations from the equilibrium point. The greater the deviation from the equilibrium, the less accurate the linearized approximation of the dynamics

This section applies the linear state feedback regulation approach. I used that approach in a previous chapter, in which the cost function in Eq. 9.3, repeated here,

$$\mathcal{J} = \int_0^T \left(\mathbf{u}'\mathbf{R}\mathbf{u} + \mathbf{x}'\mathbf{Q}\mathbf{x} \right) \mathrm{d}t,$$

balances the tradeoff between the cost of control inputs and the cost of state deviation from equilibrium. The model is written so that the equilibrium states are $\mathbf{x}^* = \mathbf{0}$. We obtain the optimal state feedback by applying the methods described in the prior chapter (see also the supplemental Mathematica code).

Consider the linear approximation in Eq. 10.2. That system has one input, for which we let $\mathbf{R} = 1$ and scale the state costs accordingly. For each state, assume that the cost is ρ^2, so that the integrand of the cost becomes $u^2 + \rho^2(x_1^2 + x_2^2)$.

We can calculate the feedback input that minimizes the cost for the linearized approximation. Using the optimal feedback, we can form a closed-loop system for both the linearized system and the original nonlinear system.

Figure 10.2 shows the response to an impulse perturbation for the closed-loop systems. In each panel, the nonlinear (blue) and linear (gold) responses are similar, showing that the design for the linear system works well for the nonlinear system.

The panels from left to right show a decreasing cost weighting on the inputs relative to the states. As the relative input costs become less heavily weighted, the optimal feedback uses stronger inputs to regulate the response, driving the system back to equilibrium more quickly.

Minimizing a cost function by state feedback may lead to systems that become unstable with respect to variations in the model dynamics. Previous chapters discussed alternative robust techniques, including integral control and combinations of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ methods. We may apply those alternative methods to the linearized approximation in Eq. 10.2. The linearized system corresponds to the transfer function

$$P(s) = \frac{n/4}{s^2 + (1 + \gamma)s + \gamma}.$$
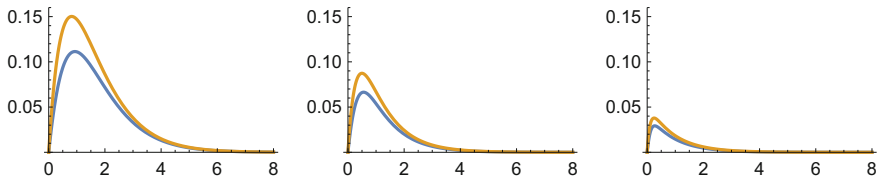
Fig. 10.2 State feedback increases the rate of return to equilibrium for the original nonlinear system (blue) and a linear approximation (gold). The optimal state feedback was calculated from the linearized system in Eq. 10.2. Each panel shows the closed-loop response to perturbation using state feedback. The perturbation is an impulse function with weighting of one, corresponding to the right panel of Fig. 10.1. The weightings of the state deviations in the cost function are, from left to right, $\rho^2 = 1, 10, 100$

Robust feedback based on this transfer function may be applied to the original nonlinear system.

10.3   Piecewise Linear Analysis and Gain Scheduling

Linear approximations at a particular operating point provide nearly exact descriptions of nonlinear dynamics near the operating point. As the system moves further from the operating point, the linear approximation becomes less accurate.

In some cases, significant divergence from the operating point causes the qualitative nature of the nonlinear dynamics to differ from the linear approximation. In other cases, such as in Fig. 10.1, the qualitative dynamics remain the same, but the quantitative responses differ.

The distance from an operating point at which the linear approximation breaks down depends on the particular nonlinear system. By considering the region over which the linear approximation holds, one can approximate a nonlinear system by a sequence of linear approximations.

Starting from an initial operating point, the first linear approximation holds near that point. Then, as the approximation breaks down away from the initial operating point, one can use a new approximation around a second operating point.

By repeatedly updating the approximation as needed for new regions, the series of linear approximations describes the nonlinear system. Each linear approximation holds in its own region or "piece." That approach leads to the piecewise linear approximation method (Rantzer and Johansson 2000).

For each piece, linear methods specify the design of feedback control. The overall control becomes a sequence of individual controllers based on linear analysis, with each particular control regime applied when the system is in the associated operating region. Alternative control in different operating regions is often called *gain scheduling*.

10.4 Feedback Linearization

Consider the simple nonlinear system with an input

$$\dot{x} = x^2 + u,$$

in which the output is equal to the single state, $y = x$ (Khalil 2002, p. 473). The equilibrium $x^* = 0$ is unstable because any perturbation from the equilibrium leads to uncontrolled growth.

The error deviation from equilibrium is x. Classical negative linear feedback would apply the control input $u = -kx$, in which the feedback is weighted by the gain, k. The closed-loop system becomes

$$\dot{x} = -kx + x^2.$$

This system has a locally stable equilibrium at zero and an unstable equilibrium at k. For a perturbation that leaves $x < k$, the system returns to its stable equilibrium. For a perturbation that pushes x beyond k, the system grows without bound. Thus, linear feedback provides local stability. The stronger the feedback, with larger k, the broader the local region of stability.

In this case, linear feedback transforms an unstable open-loop system into a locally stable closed-loop system. However, the closed-loop system remains nonlinear and prone to instability.

If we choose feedback to cancel the nonlinearity, $u = -kx - x^2$, then we obtain the linearly stable closed-loop system, $\dot{x} = -kx$.

Once we have a linear closed-loop system, we can treat that system as a linear open-loop subsystem, and use linear techniques to design controllers and feedback to achieve performance goals.

For example, we could consider the feedback linearized dynamics as $\dot{x} = -kx + v$, in which v is an input into this new linearized subsystem. We could then design feedback control through the input v to achieve various performance goals, such as improved regulation to disturbance or improved tracking of an input signal.

A nonlinear system can be linearized by feedback if the states can be written in the form

$$\dot{x} = f(x) + g(x)u. \tag{10.3}$$

Such systems are called *input linear*, because the dynamics are linear in the input, u. These systems are also called *affine in input*, because a transformation of the form $a + bu$ is an affine transformation of the input, u. Here, f and g may be nonlinear functions of x, but do not depend on u.

In the example $\dot{x} = x^2 + u$, we easily found the required feedback to cancel the nonlinearity. For more complex nonlinearities, geometric techniques have been developed to find the linearizing feedback (Slotine and Li 1991; Isidori 1995;

Khalil 2002). Once the linearized system is obtained, one may apply linear design and analysis techniques to study or to alter the system dynamics.

Feedback linearization depends on an accurate model of the dynamics. For example, if the actual model is

$$\dot{x} = ax^2 + u,$$

and the feedback linearization is taken as $u = -kx - x^2$ under the assumption that $a = 1$, then the closed-loop system is

$$\dot{x} = -kx + (a - 1)x^2.$$

If the true value of the parameter is $a = 2$, then the feedback system has dynamics $\dot{x} = -kx + x^2$, which is unstable for $x > k$.

This example shows that feedback linearization is not robust to model uncertainties. The following chapter discusses an alternative method that can provide robust feedback control for nonlinear systems.

Chapter 11
Adaptive Control

The parameters of a process may be unknown or may change slowly over time. How can one control a process with unknown parameters?

Earlier chapters discussed robust methods. Those methods provide good response to a broad set of alternative process dynamics.

This chapter presents adaptive control, in which the control system adjusts itself by using measurements of the system's response. I follow the example from Sect. 6.2.4 of Ioannou and Sun (2012).

In adaptive control, the system repeatedly updates the controller parameters to reduce the error between the system's actual output and the output of an ideal target response model. Figure 11.1 shows the structure of a common approach known as *model reference adaptive control.*

Suppose the process dynamics are given by the affine form in Eq. 10.3 as

$$\dot{y} = af(y) + bg(y)u, \tag{11.1}$$

which describes linear systems and also a wide variety of nonlinear systems. In this example, we know the functions f and g, but do not know the parameter values for a and b. The goal is to design a control input, u, that causes the system output, y, to match the output of a specified model.

11.1 General Model

Typically, one chooses a simple linear model for the design target. In this example, we use

$$\dot{y}_m = -a_m y_m + b_m r. \tag{11.2}$$

Here, the parameters a_m and b_m are known aspects of the target model specification, and r is the reference or external environmental input. For a constant reference input, this model converges to the reference exponentially at rate a_m, with amplitude
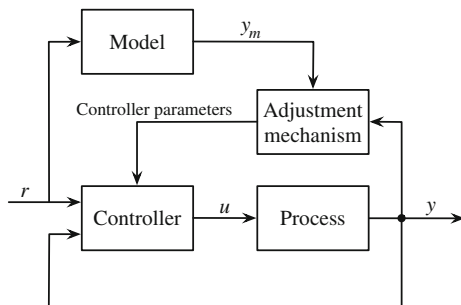
Fig. 11.1 Model reference adaptive control. The goal is to construct a controller so that the system output, y, matches the output of a specified model, y_m. To achieve that goal, the lower feedback loop with controller and process must together form a system that has the same dynamics as the model. If parameters of the process are unknown, one can use measurement of the error, $e = y - y_m$, to adaptively adjust the parameters of the controller in response to the error. Ideally, the system learns controller parameters such that the output, y, converges to match the target model output, y_m. Redrawn from Fig. 5.1 of Åström and Wittenmark (2008), © Karl J. Åström and Björn Wittenmark

of the response relative to the input of b_m/a_m. Figure 11.2 illustrates the design target response for a sinusoidal input, r.

For given values of a and b, the control input

$$u = \frac{1}{g(y)}\big[k_1^* f(y) + k_2^* y + w^* r\big]$$

$$k_1^* = -\frac{a}{b} \qquad k_2^* = -\frac{a_m}{b} \qquad w^* = \frac{b_m}{b} \tag{11.3}$$

transforms the process model in Eq. 11.1 into the target model in Eq. 11.2.

If the parameters a and b are unknown, then the input, u, must be based on the estimates for $k_1(t)$, $k_2(t)$, and $w(t)$. The estimates are updated by an adaptive process in response to the error difference between system and model output, $e = y - y_m$. The dynamics of the error are $\dot{e} = \dot{y} - \dot{y}_m$.

To obtain an expression for $\dot{e}$, we need a modified form of $\dot{y}$ that contains only the known parameters a_m and b_m and the estimates k_1, k_2, and w. The first step expresses the process dynamics in Eq. 11.1 by adding and subtracting $b\big[k_1^* f(y) + k_2^* y + w^* r\big]$ and using the identities $bk_1^* = -a$ and $bk_2^* = -a_m$ and $bw^* = b_m$, yielding

$$\dot{y} = -a_m y + b_m r$$
$$+ b\big[-k_1^* f(y) - k_2^* y - w^* r + u g(y)\big].$$

Write the tracking errors as $\tilde{k}_1 = k_1 - k_1^*$ and $\tilde{k}_2 = k_2 - k_2^*$ and $\tilde{w} = w - w^*$. The error dynamics can then be written as
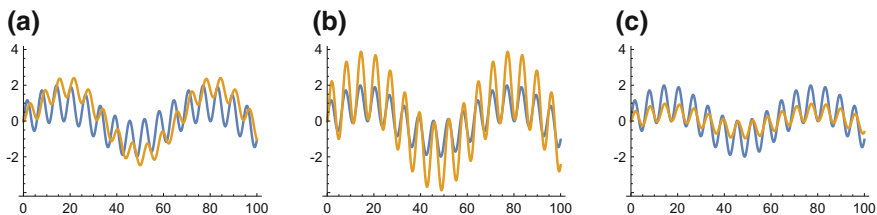
Fig. 11.2 Illustrations of the model response (gold curve) in Eq. 11.2 for reference input (blue curve) given by $r(t) = \sin(t/10) + \sin(t)$. The speed of the tracking response increases with the parameter a_m, and the amplitude of the response increases with b_m/a_m. **a** A slow, lagging response with $a_m = 0.3$ and a twofold amplitude enhancement with $b_m/a_m = 2$. **b** A fast tracking response with $a_m = 3$ and a twofold amplitude enhancement with $b_m/a_m = 2$. **c** A fast tracking response with $a_m = 3$ and a twofold amplitude reduction with $b_m/a_m = 1/2$

$$\dot{e} = -a_m e + b\left[\tilde{k}_1 f(y) + \tilde{k}_2 y + \tilde{w} r\right].$$

To analyze the error dynamics, we need expressions for the processes used to update the parameter estimates. A common choice is

$$\dot{k}_1 = -\gamma_1 e f(y)$$
$$\dot{k}_2 = -\gamma_2 e y$$
$$\dot{w} = -\gamma_3 e r,$$

in which I have assumed that $b > 0$.

11.2 Example of Nonlinear Process Dynamics

The general results of the prior section can be applied to any linear process or to any nonlinear process that can be approximated by the affine form of Eq. 11.1. For this nonlinear example, let

$$\dot{y} = ay^2 + bu, \tag{11.4}$$

with $f(y) = y^2$ and $g(y) = 1$.

Figure 11.3 illustrates the rate of adaptation for various parameters. As the adaptation parameters, γ, increase, the system output converges increasingly rapidly to the target model output.
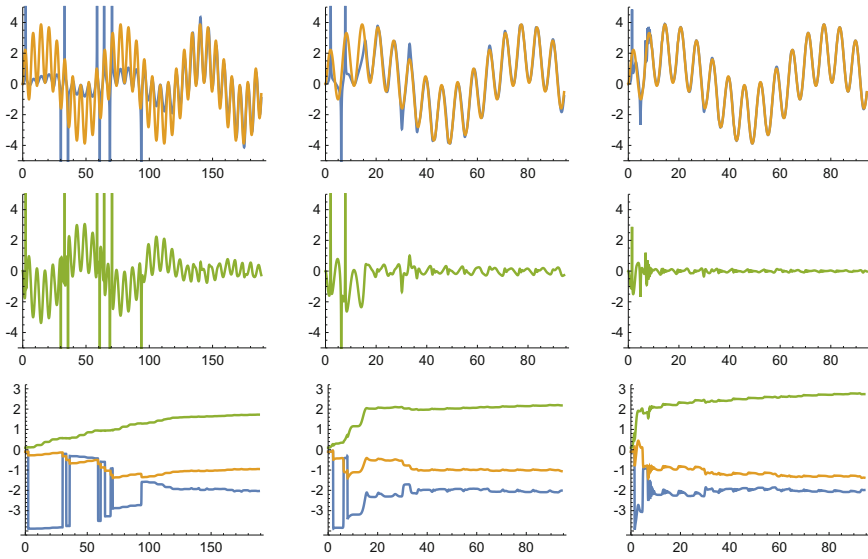
Fig. 11.3 Rate of adaptation for the parameters that set u, the control input into the system process, as given in Eq. 11.3. The controller parameters adapt so that the system output tracks the output of a reference model that expresses the design goal. The equations in the text describe the system and adaptation dynamics. The top row shows the system's output in blue and the target model's output in gold. The middle row is the system's error, which is the difference between the system and model responses in the upper row. The bottom row shows the adaptation of the control parameters k_1, k_2, and w, in blue, gold, and green, respectively. This figure uses the reference model parameters $a_m = 3$, $b_m = 6$, and the true underlying system process parameters, $a = 4$, and $b = 2$. The controller parameters that adapt are k_1, k_2, and w, which are the system's estimates for the underlying process parameters in relation to the model parameters. The rate of adaptation is set by $\gamma_1 = \gamma_2 = \gamma_3$, which are, for the columns from left to right, $0.01, 0.1, 1$, respectively. The faster the rate of adaptation, the more quickly the system converges to the reference model

11.3   Unknown Process Dynamics

The previous section assumed a particular form for the process dynamics in Eq. 11.4, with unknown parameters a and b. How could we handle a process with unknown dynamics?

   One simple approach is to assume a very general form for the process dynamics, such as a polynomial

$$\dot{y} = a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n + bu,$$

and then run the adaptation process on the parameters $(a_0, a_1, \ldots, a_n, b)$. One could use other generic forms for the dynamics and estimate the parameters accordingly.

This approach provides a way for the system output to mimic the model output, without the system necessarily converging to use the same mathematical description of dynamics as in the model.

Chapter 12
Model Predictive Control

Control design often seeks the best trajectory along which to move a system from its current state to a target state. Most control methods approximate this goal by using the current inputs and system state to calculate the next control signal to drive system dynamics. That standard approach considers only the first step of the full trajectory toward the target state. The idea is that estimating a good first step in the right direction is sufficient, without consideration of the full trajectory from the current location to the final target.

Model predictive control considers the full sequence of input steps required to move the system optimally from its current state to a future target. The control system then applies the first inputs to start the system along that optimal trajectory (Rossiter 2004; Camacho and Bordons 2007; Ellis et al. 2014; Rawlings and Mayne 2015).

After applying the initial inputs, the system does not use the additional sequence of calculated inputs to continue along the planned trajectory. Instead, the system takes updated measures of the external target and the internal state. The new information is used to recalculate an updated optimal trajectory. Using the updated trajectory, the newly calculated first inputs are then applied to the system. The process repeats with each new round of updated external and internal signals.

This approach considers a receding future horizon. At each point in time, the system calculates the optimal trajectory to a particular time point in the future—the horizon. Then, after a small amount of time passes relative to the future horizon, the system recalculates by taking current inputs and advancing the future horizon by the time elapsed.

Intuitively, this approach seems similar to many decisions made by humans. We estimate how we will get to a goal, start off in the best direction, then update our plan as new information arrives.

Our estimate of how we will get to a goal depends on an internal model of our dynamics and on the modulating control signals that we will use to alter our dynamics. The self-correcting process of recalculating the planned trajectory means that we do not need an accurate model of our internal dynamics to perform well. An approximate or misspecified model of dynamics often works well, even for nonlinear processes.

Model predictive control may be enhanced by adaptive feedback that modifies the parameters or the form for the model of internal dynamics. The general approaches of model predictive control and adaptive control provide benefits of robustness with respect to the model of internal dynamics.

12.1   Tracking a Chaotic Reference

Figure 12.1 shows the performance of a simple model predictive control system when tracking a chaotic reference signal. The figure caption describes the calculation of the chaotic input signal (blue curve). In this example, the model predictive control system begins with an internal process given by

$$\ddot{x} = u \tag{12.1}$$

for control input, u. The input can be thought of as a force that alters the acceleration of the system state.

In this example, the cost function is summed over a series of discrete time points from the present until a future horizon at time T, yielding

$$\mathcal{J} = \sum_{t=0}^{T} \left[x(t) - r(t) \right]^2 + \rho u(t)^2.$$

The first term is the squared tracking error. The second term is the squared control input signal, weighted by ρ, which is the relative cost of the input signal to the tracking error. The sum includes a series of input controls and tracking deviations over a finite period from the present to the future horizon, T. As time passes, the time window $(0, T)$ moves so that $t = 0$ is always the present time.

The system has sensors to measure the current locations and time derivatives (velocities) for the system state and the reference signal, given as x_0, $\dot{x}_0$, r_0, and
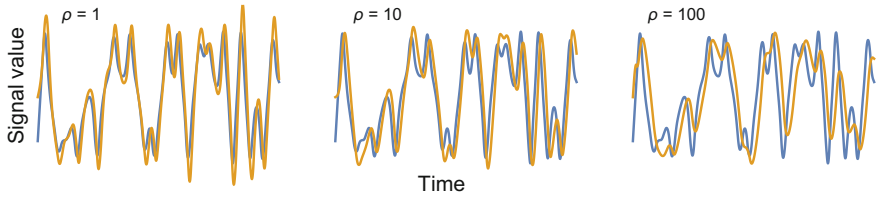
Fig. 12.1 Tracking an environmental reference signal by model predictive control. The blue curves show the environmental signal, a chaotic form of Duffing's equation, $\ddot{r} + 0.15\dot{r} - r + r^3 = 0.3\cos(t)$, with initial conditions $\dot{r} = 1.001$ and $r = -1$. The gold curves show the tracking performance for different values for the cost, ρ, of control input signals relative to the tracking error. A higher control input cost favors weaker control inputs and greater tracking error

$\dot{r}_0$. From those measurements, and from the projected set of inputs, $u(t)$, over the interval $t = 0, 1, \ldots, T$, the system can project the values of $x(t)$ and $r(t)$ and thus estimate the tracking errors and the total cost, $\mathcal{J}$. In particular,

$$x(t) = x_0 + \dot{x}_0 t + \sum_{\alpha=0}^{t-1} (t - \alpha) u(\alpha)$$

$$r(t) = r_0 + \dot{r}_0 t.$$

Here, $u(\alpha)$ is interpreted as an impulse that acts at the continuous time offset, α, relative to the current time. Because the system has dynamics $\ddot{x} = u$, an impulse at α causes an instantaneous increase in velocity by $u(\alpha)$, which then acts to change the future predicted value at time t by $(t - \alpha) u(\alpha)$.

These predicted values for x and r allow calculation of the sequence of control inputs $u(t)$ over the interval $t = 0, 1, \ldots, T$ that minimize the cost, $\mathcal{J}$.

The plots in Fig. 12.1 show the system trajectory that reflects the minimization of $\mathcal{J}$. At each time step, the system calculates the sequence u to minimize $\mathcal{J}$, then applies $u(0)$ as the control input. The remaining u values for $t = 1, 2, \ldots, T$ are ignored. In the next time step, the same procedure gives the new control input for that time period, and the future inputs for the optimal trajectory are again ignored. The process continues for each time period as the future horizon recedes.

12.2 Quick Calculation Heuristics

The solution for the input $u(0)$ typically depends on the full sequence of inputs over $t = 0, 1, \ldots, T$. In some cases, a relatively simple explicit solution for $u(0)$ can be obtained that requires only the current measured inputs for x_0, $\dot{x}_0$, r_0, and $\dot{r}_0$.

If a system applies only the first input, $u(0)$, before recalculating in the next time step, then that system only needs the explicit solution for $u(0)$ to update the control inputs in each time step.

For example, with $T = 2$, the exact solution for the above case is

$$u(0) = \frac{\Delta + \rho[3\Delta + 2(\dot{r}_0 - \dot{x}_0)]}{1 + 6\rho + \rho^2},$$

with $\Delta = (r_0 + \dot{r}_0) - (x_0 + \dot{x}_0)$. With larger T, the solution has more terms as powers of ρ, but nonetheless remains a relatively simple ratio of polynomials in ρ that could be approximated by a quick heuristic calculation.

12.3   Mixed Feedforward and Feedback

In some cases, the sensor and calculation costs of updating in each time step may not provide sufficient benefit. Instead, the system could apply the first few control inputs of the sequence, $u(0), u(1), \ldots, u(\tau)$, and then update the sequence at time $\tau < T$.

A system that uses feedback inputs at one time point to calculate and then apply a future sequence of control inputs is running in partial feedforward mode. The feedback inputs arrive, and then the system runs forward from those inputs without the feedback correction obtained by comparing the changing system output to the potentially changing target reference signal.

After a while, the system may take new input readings and update the projected sequence of future control signals. Each measurement and recalculation acts as a feedback correction process. Thus, systems may combine the simplicity and relatively low cost of feedforward control with the correction and robustness benefits of feedback.

12.4   Nonlinearity or Unknown Parameters

This section's example used a simple model of internal dynamics, $\ddot{x} = u$, given in Eq. 12.1. That expression, equating acceleration and force, provided a simple way in which to analyze trajectories. That internal model may often perform well even if the true model is nonlinear because the first move along the calculated trajectory often depends on how the force of the applied input alters the acceleration of the system.

Alternatively, one could use a more general expression for the internal model dynamics, with a set of unknown parameters. One could then add an adaptive control layer to the system to provide updated parameter estimates. In some cases, this combination of model predictive control and adaptive control may perform well.

Chapter 13
Time Delays

You suddenly notice a ball flying toward your head. Your first reaction happens after a delay. To avoid the ball, you must consider where your head will be after its delayed response in relation to where the ball will be.

This chapter presents models for delay dynamics and discusses a control method that compensates for delays.

13.1 Background

Delays often occur in the signals that flow between components of a control system. An uncompensated delay may reduce system performance. Suppose, for example, that the sensor measuring the system output, y, requires δ time units to process and pass on its measured value as a feedback signal.

The delayed feedback signal reports the system output δ time units before the current time, which we write as $y(t - \delta)$. The calculated error between the current reference input and the delayed feedback, $r(t) - y(t - \delta)$, may not accurately reflect the true error between the target value and the current system output value, $r(t) - y(t)$.

Delays may destabilize a system. If the calculated error overestimates the true error, then the system may overcompensate, pushing the system output away from the target reference value rather than toward it.

The robust control methods discussed in earlier chapters can reduce the instabilities created by delays. Robust control creates a significant stability margin. A large stability margin means that factors not directly included in the design, such as unknown delays, will usually not destabilize the system.

(a)

(b)
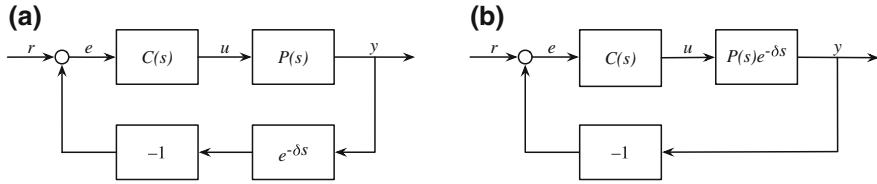
Fig. 13.1 Time delays in feedback loops. **a** Sensor delay. The sensor that measures system output and passes that value as feedback has a delay of δ time units between the system input and the measured output. The transfer function $e^{-\delta s}$ passes its input unmodified but with a delay of δ time units. **b** Process delay. The system process, $Pe^{-\delta s}$, has a lag of δ time units between the time at which a control input signal, u, is received and the associated system output signal, y, is produced

In addition to general robust approaches, many specific design methods deal explicitly with delays. The delays are often called *dead time* or *transport lag* (Åström and Hägglund 2006; Normey-Rico and Camacho 2007; Visioli and Zhong 2011).

The design methods typically use a prediction model. A prediction allows the system to use measured signal values at time $t - \delta$ to estimate the signal values at time t.

13.2   Sensor Delay

Figure 13.1a shows a standard feedback loop with a sensor delay. The sensor that measures the process output, y, delays passing on the measured value by δ time units.

In Fig. 13.1a, the transfer function $e^{-\delta s}$ describes the delay. That transfer function passes its input unmodified, but with a delay of δ. Thus, the measured output that is passed by the sensor as feedback is given by the transfer function $Ye^{-\delta s}$, which transforms inputs, $y(t)$, into the time-delayed outputs, $y(t - \delta)$.

We can derive how the delay influences the closed-loop system response in Fig. 13.1a. Define the open loop of the system as $L = CP$, as in Eq. 3.4. Then we can write the system output as $Y = LE$, the error input, E, multiplied by the open-loop system process, L.

The error is the difference between the reference input and the feedback output from the sensor, $E = R - Ye^{-\delta s}$. Substituting this expression for the error into $Y = LE$, we obtain the transfer function expression for the closed-loop system response, $G = Y/R$, as

$$G(s) = \frac{L(s)}{1 + L(s)e^{-\delta s}}. \tag{13.1}$$

13.3   Process Delay

Figure 13.1b illustrates a feedback system with a process delay. The full process, $Pe^{-\delta s}$, requires δ time units to transform its input to its output. Thus, the process output lags behind the associated control input to the process by δ time units.

The open loop in Fig. 13.1b is $Le^{-\delta s} = CPe^{-\delta s}$. We can derive the closed-loop system response by the method used to derive Eqs. 3.4 and 13.1, yielding

$$G(s) = \frac{L(s)e^{-\delta s}}{1 + L(s)e^{-\delta s}}. \tag{13.2}$$

The simple transfer function description for signal delays allows one to trace the consequences of delays through a system with many components that are each approximately linear.

13.4   Delays Destabilize Simple Exponential Decay

This section illustrates how delays can destabilize a system. I analyze a simple open-loop integrator, $L(s) = k/s$. That transfer function corresponds to dynamics given by $\dot{x}(t) = kr(t)$, for reference input r, which has solution $x(t) = k \int_0^t r(\tau)d\tau$ for initial condition $x_0 = 0$. Thus, the output of L is the integral of its input multiplied by the gain, k. I assume throughout this section that the output equals the system state, $y(t) = x(t)$.

A standard negative feedback system has transfer function $G = L/(1 + L)$, which for $L = k/s$ is

$$G(s) = \frac{k}{k + s},$$

which has dynamics

$$\dot{x}(t) = -kx(t) + kr(t) = k[r(t) - x(t)].$$

The error signal is $r(t) - x(t)$. The solution is the integral of the error signal.

For constant input, $\hat{r} = r(t)$, the solution is a constant exponential decay toward the equilibrium setpoint at rate k. Without loss of generality, we can take the setpoint as $\hat{r} = 0$ and write the solution as

$$x(t) = x_0 e^{-kt}.$$

We can apply the same approach for the sensor delay system in Eq. 13.1. For $L = k/s$, the system transfer function is
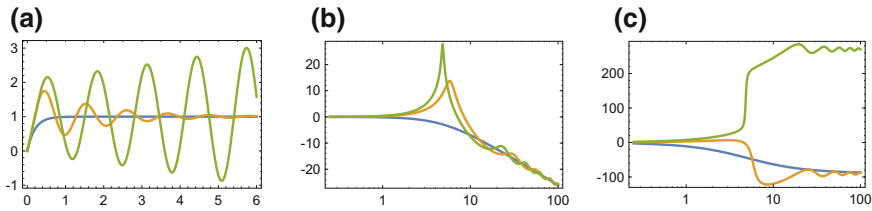
Fig. 13.2 Feedback delay destabilizes a simple integrator process. **a** Temporal dynamics from Eq. 13.4, with gain $k = 5$ and unit step input $r(t) = 1$. The feedback delays are $\delta = 0, 0.25, 0.33$ shown in the blue, gold, and green curves, respectively. **b** Bode gain plot of the associated transfer function in Eq. 13.3. Greater feedback lag increases the resonant peak. **c** Bode phase plot. Note how the destabilizing feedback lag (green curve) creates a large phase lag in the frequency response

$$G(s) = \frac{k}{ke^{-\delta s} + s},\tag{13.3}$$

in which the term $e^{-\delta s}$ expresses the delay by δ. The differential equation for this system is

$$\dot{x}(t) = k[r(t) - x(t - \delta)],\tag{13.4}$$

which, for reference input $\hat{r} = 0$, is

$$\dot{x}(t) = -kx(t - \delta).$$

This system expresses a delay differential process. Although this delay differential system is very simple in structure, there is no general solution. A sufficiently large delay, δ, destabilizes the system because the rate of change toward the equilibrium setpoint remains too high when that rate depends on a past value of the system state.

In particular, the dynamics in Eq. 13.4 describe a simple lagged feedback system. At each time, t, the error between the target value and the system state from δ time units ago is $\hat{r} - x(t - \delta)$. That lagged error, multiplied by the feedback gain, k, sets the rate at which the system moves toward the setpoint.

Because the system state used for the feedback calculation comes from a lagged time period, the feedback may not accurately reflect the true system error at time t. That miscalculation can destabilize the system.

Figure 13.2a shows how feedback lag can destabilize simple exponential decay toward an equilibrium setpoint. With no time lag, the blue curve moves smoothly and exponentially toward the setpoint. The gold curve illustrates how a relatively small feedback lag causes this system to move toward the setpoint with damped oscillations. The green curve shows how a larger feedback lag destabilizes the system. The Bode plots in Fig. 13.2b, c illustrate how feedback delay alters the frequency and phase response of the system in destabilizing ways.
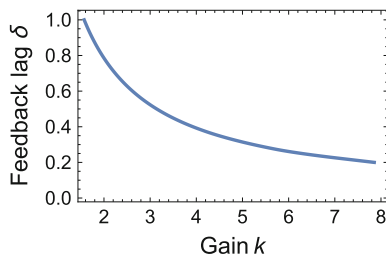
Fig. 13.3 Greater process gain, k, can be destabilized by smaller feedback lag, δ. Combinations of gain and lag below the curve are stable. Combinations above the curve are unstable. Stability is determined by the maximum real part of the eigenvalues for Eq. 13.4 with constant reference input

In earlier chapters, I showed that high gain feedback systems move rapidly toward their setpoint but may suffer sensitivity to destabilizing perturbations or uncertainties. Feedback lag may be thought of as a kind of perturbation or uncertainty.

Figure 13.3 shows how the system gain, k, enhances the destabilizing effect of feedback lag, δ. Combinations of gain and lag below the curve are stable. Combinations above the line are unstable. Systems with greater gain can be destabilized by smaller feedback lag.

Process delays differ from feedback delays only in the extra lag associated with the reference input. For the process delay system given by the transfer function in Eq. 13.2, the dynamics are

$$\dot{x}(t) = k[r(t - \delta) - x(t - \delta)],$$

which describe an error integrator lagged by $t - \delta$. For constant reference input, $r(t) = \hat{r}$, the process delay dynamics are the same as for the feedback delay dynamics in Eq. 13.4.

13.5 Smith Predictor

Compensating for a time delay requires prediction. Suppose, for example, that there is a process delay between input and output, as in Fig. 13.1b. The Smith predictor provides one way to compensate for the delay. To understand the Smith predictor, we first review the process delay problem and how we might solve it.

In Fig. 13.1b, the time-delay transfer function in the process, $e^{-\delta s}$, maps an input signal at time t to an output that is the input signal at $t - \delta$. Thus, the open loop $CPe^{-\delta s}$ transforms the current input, $r(t)$, to the output, $y(t - \delta)$. The measured error between input and output, $r(t) - y(t - \delta)$, gives an incorrect signal for the feedback required to push the tracking error, $r(t) - y(t)$, toward zero.

One way to obtain an accurate measure of the tracking error is to predict the output, $y(t)$, caused by the current input, $r(t)$. The true system process, $Pe^{-\delta s}$, has
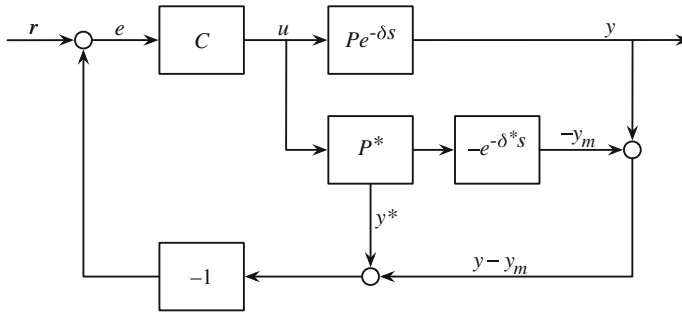
Fig. 13.4 Smith predictor to compensate for time delay in the process output. Redrawn from Fig. 5.1 of Normey-Rico and Camacho (2007), © Springer-Verlag

a lag, and the unlagged process, P, may be unknown. If we could model the way in which the process would act without a lag, P^*, then we could generate an estimate, $y^*(t)$, to predict the output, $y(t)$.

Figure 13.4 shows the feedback pathway through P^*. If P^* is an accurate model of P, then the feedback through P^* should provide a good estimate of the tracking error. However, our goal is to control the actual output, y, rather than to consider output estimates and feedback accuracy. The Smith predictor control design in Fig. 13.4 provides additional feedbacks that correct for potential errors in our model of the process, P^*, and in our model of the delay, δ^*.

In Fig. 13.4, the pathway through P^* and then $e^{\delta^* s}$ provides our model estimate, y_m, of the actual output, y. The error between the true output and the model output, $y - y_m$, is added to the estimated output, y^*, to provide the value fed back into the system to calculate the error. By using both the estimated output and the modeling error in the feedback, the system can potentially correct discrepancies between the model and the actual process.

The system transfer function clarifies the components of the Smith predictor system. The system transfer function is $G = Y/R$, from input, R, to output, Y. We can write the system transfer function of the Smith predictor in Fig. 13.4 as

$$G = \left(\frac{CP}{1 + C\,(P^* + \Delta M)} \right) e^{-\delta s}, \tag{13.5}$$

in which the modeling error is

$$\Delta M = Pe^{-\delta s} - P^* e^{-\delta^* s}.$$

The *Derivation* at the end of this chapter shows the steps to Eq. 13.5.

The stability of a transfer function system depends on the form of the denominator. In the case of Eq. 13.5, the eigenvalues are the roots of s obtained from $1 + C(P^* + \Delta M) = 0$. We know the process, P^*, because that is our model to estimate the unknown system, P.

To obtain robust stability, we can design a controller, C, under the assumption that the modeling error is zero, $\Delta M = 0$. For example, we can use the methods from the earlier chapter *Stabilization* to obtain a good stability margin for C relative to P^*. Then we can explicitly analyze the set of modeling errors, ΔM, for which our robust controller will remain stable. A design with a good stability margin also typically provides good performance.

13.6   Derivation of the Smith Predictor

The derivation of Eq. 13.5 begins with the transfer functions obtained directly from Fig. 13.4 for various outputs

$$Y = ECPe^{-\delta s}$$

$$Y^* = ECP^* = Y\frac{P^*}{Pe^{-\delta s}}$$

$$Y_m = ECP^* e^{-\delta^* s} = Y\frac{P^* e^{-\delta^* s}}{Pe^{-\delta s}}$$

with error input

$$E = R - Y - Y^* + Y_m$$

$$= R - Y\left(1 + \frac{P^*}{Pe^{-\delta s}} - \frac{P^* e^{-\delta^* s}}{Pe^{-\delta s}}\right)$$

$$= R - Y\frac{1}{Pe^{-\delta s}}\left(P^* + \Delta M\right)$$

with

$$\Delta M = Pe^{-\delta s} - P^* e^{-\delta^* s}.$$

Substituting the expression for E into the expression for Y yields

$$Y = CPe^{-\delta s}[R - Y\frac{1}{Pe^{-\delta s}}\left(P^* + \Delta M\right)].$$

The system response, Y, to an input, R, is $G = Y/R$, which we obtain by dividing both sides of the prior equation by R, yielding

$$G = CPe^{-\delta s} - GC\left(P^* + \Delta M\right),$$

from which we obtain

$$G = \left(\frac{CP}{1 + C\left(P^* + \Delta M\right)}\right)e^{-\delta s},$$

which matches Eq. 13.5. When the model is accurate, $P = P^*$ and $\Delta M = 0$, the system reduces to

$$G = \left(\frac{CP^*}{1 + CP^*} \right) e^{-\delta s}$$

for known model P^*. This transfer function has the standard form of a negative feedback system with open loop $L = CP^*$.

Chapter 14
Summary

Many other control approaches and applications have been developed (Baillieul and Samad 2015). Those extensions build on the foundational principles emphasized in this tutorial. Three key principles recur.

14.1 Feedback

> There are two, and only two, reasons for using feedback. The first is to reduce the effect of any unmeasured disturbances acting on the system. The second is to reduce the effect of any uncertainty about systems dynamics.
>
> —Vinnicombe (2001, p. xvii)

Feedback is unnecessary if one has a complete, accurate model of system dynamics. With an accurate model, one can map any input to the desired output. A direct feedforward open loop does the job.

However, unpredictable perturbations occur. Models of dynamics almost always incorrectly specify the true underlying process.

Correcting errors by feedback provides the single most powerful design method. Natural systems that control biological function often use feedback. Human-engineered systems typically correct errors through feedback.

14.2 Robust Control

> [H]ow much do we need to know about a system in order to design a feedback compensator that leaves the closed loop behaviour insensitive to that which we don't know?
>
> —Vinnicombe (2001, p. xvii)

Robustness means reduced sensitivity to disturbance or modeling error. Feedback improves robustness. However, feedback only describes a broad approach.

Many specific methods refine the deployment of feedback. For example, filters reduce the resonant peaks in system response. Controllers modulate dynamics to improve stability margin.

A large stability margin means that the system can maintain stability even if the true process dynamics depart significantly from the simple linear model used to describe the dynamics.

14.3 Design Tradeoffs and Optimization

A well-performing system moves rapidly toward the desired setpoint. However, rapid response can reduce stability. For example, a strong response to error can cause a system to overshoot its setpoint. If each overshoot increases the error, then the system diverges from the target.

The fast response of a high-performing system may destabilize the system or make it more sensitive to disturbances. A tradeoff occurs between performance and robustness.

Many other tradeoffs occur. For example, control signals modulate system dynamics. The energy required to produce control signals may be expensive. The costs of control signals trade off against the benefits of modulating the system response.

The sensitivity of a system to perturbations varies with the frequency at which the signal disturbs the system. Often, reduced sensitivity to one set of frequencies raises sensitivity to another set of frequencies.

Optimization provides a rigorous design approach to tradeoffs. One may assign costs and benefits to various aspects of performance and robustness or to the response at different frequencies. One can then consider how changes in system design alter the total balance of the various costs and benefits. Ideally, one finds the optimal balance.

14.4 Future Directions

Control theory remains a very active subject (Baillieul and Samad 2015). Methods such as robust $\mathcal{H}_\infty$ analysis and model predictive control are recent developments.

Computational neural networks have been discussed for several decades as a method for the control of systems (Antsaklis 1990). Computational networks are loosely modeled after biological neural networks. A set of nodes takes inputs from the environment. Each input node connects to another set of nodes. Each of those intermediate nodes combines its inputs to produce an output that connects to yet another set of nodes, and so on. The final nodes classify the environmental state, possibly taking action based on that classification (Nielsen 2015; Goodfellow et al. 2016).

For many years, neural networks seemed like a promising approach for control design and for many other applications. However, that approach typically faced various practical challenges in implementation. Until recently, the practical problems meant that other methods often worked better in applications.

New methods and increased computational power have made neural networks the most promising approach for major advances in control system design. Spectacular examples include self-driving cars, real-time computer translation between languages, and the reshaping of modern financial markets. At a simpler level, we may soon see many of the control systems in basic daily devices driven by embedded neural networks instead of the traditional kinds of controllers.

The rise of neural networks also foreshadows a potential convergence between our understanding of human-designed engineering systems and naturally designed biological systems (Frank 2017).

In a human-designed system, an engineer may build a controller to improve the total benefits that arise from tradeoffs between cost, performance, and robustness. In biology, natural selection tends to build biochemical or physical systems that improve the tradeoffs between various dimensions of biological success. Those biological dimensions of success often can be expressed in terms of cost, performance, and robustness.

The similarities and differences between human-designed systems and naturally designed systems will provide many insights in the coming years. An understanding of the basic concepts of control design will be required to follow future progress and to contribute to that progress.

References

Anderson BDO, Moore JB (1989) Optimal control: linear quadratic methods. Prentice Hall, Englewood Cliffs, New Jersey

Antsaklis PJ (1990) Neural networks in control systems. IEEE Control Syst Mag 10:3–5

Astolfi A, Karagiannis D, Ortega R (2008) Nonlinear and adaptive control with applications. Springer, New York

Åström KJ, Hägglund T (2006) Advanced PID control. Systems and Automation Society, Research Triangle Park, North Carolina, USA, ISA-The Instrumentation

Åström KJ, Murray RM (2008) Feedback systems: an introduction for scientists and engineers, version v2.11a edn. Princeton University Press, Princeton, NJ

Åström KJ, Wittenmark B (2008) Adaptive control, 2nd edn. Dover Publications, Mineola

Baillieul J, Samad T (eds) (2015) Encyclopedia of systems and control. Springer-Verlag, London

Bode HW (1964) Feedback—the history of an idea. In: Bellman RE, Kalaba RE (eds) Selected papers on mathematical trends in control theory. Dover, New York, pp 106–123

Camacho EF, Bordons C (2007) Model predictive control, 2nd edn. Springer, London

Chang YC (2017) Adaptive H_2/H_∞ tracking control for a class of uncertain robotic systems. Int J Control 90:463–479

Chen X, Zhou K (2001) Multiobjective H_2/H_∞ control design. SIAM J Control Optim 40(2):628–660

Dorf RC, Bishop RH (2016) Modern control systems, 13th edn. Pearson, Santa Monica

Ellis M, Durand H, Christofides PD (2014) A tutorial review of economic model predictive control methods. J Process Control 24(8):1156–1178

Frank SA (2017) Puzzles in modern biology. V. Why are genomes overwired? F1000Research 6:924

Garpinger O, Hägglund T, Åström KJ (2014) Performance and robustness trade-offs in PID control. J Process Control 24(5):568–577

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge. http://www.deeplearningbook.org

Ioannou PA, Sun J (2012) Robust adaptive control. Dover Publications, Mineola

Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, New York

Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River

Liu KZ, Yao Y (2016) Robust control: theory and applications. Wiley, Singapore

Nielsen MA (2015) Neural networks and deep learning. Determination Press. http://neuralnetworksanddeeplearning.com/

Normey-Rico JE, Camacho EF (2007) Control of dead-time processes. Springer, London

Ogata K (2009) Modern control engineering, 5th edn. Prentice Hall, New York

Parker GA, Maynard Smith J (1990) Optimality theory in evolutionary biology. Nature 348:27–33

Qiu L, Zhou K (2010) Introduction to feedback control. Prentice Hall, Upper Saddle River, NJ

Qiu L, Zhou K (2013) Preclassical tools for postmodern control: an optimal and robust control theory for undergraduate education. IEEE Control Syst 33(4):26–38

Rantzer A, Johansson M (2000) Piecewise linear quadratic optimal control. IEEE Trans Automatic
    Control 45(4):629–637
Rawlings JB, Mayne DQ (2015) Model predictive control: theory and design. Nob Hill Publishing,
    Madison
Rossiter JA (2004) Model-based predictive control: a practical approach. CRC Press, Boca Raton
Slotine JJE, Li W (1991) Applied nonlinear control. Prentice Hall, Englewood Cliffs
Vinnicombe G (2001) Uncertainty and feedback: H_∞ loop-shaping and the ν-gap metric. Imperial
    College Press, London
Visioli A, Zhong Q (2011) Control of integral processes with dead time. Springer, London
Wolfram Research (2017) Mathematica, Version 11.2. Champaign, IL
Zhou K, Doyle JC (1998) Essentials of robust control. Prentice Hall, Upper Saddle River

Index